



**T.C. İSTANBUL T CARET
 NİVERSİTESİ**

FEN B LİMLERİ ENSTİT S 

**D NAMİK PARAMETRELİ UZAYSAL ARAMA İLE UYGUN D Ğ M
SE İMİ ALGORİTMALARININ SİM LASYONU VE
KAR ILA TIRILMASI**

Mert Can G RG N

**Dan şman
Dr.  ğr.  yesi Ali BOYACI**

**Y KSEK L SANS TEZİ
KENTSEL S STEMLER VE ULA TIRMA Y NETİMİ ANAB LİM DALI
İSTANBUL - 2020**

KABUL VE ONAY SAYFASI

Mert Can GİRGİN tarafından hazırlanan " **Dinamik Parametrelili Uzaysal Arama İle Uygun D ğ m Se imi Algoritmalarının Sim lasyonu ve Karşılaştırılması** " adlı tez  alışması 13/07/2020 tarihinde aşığıdaki j ri  yeleri  n nde başarı ile savunularak, İstanbul Ticaret  niversitesi Fen Bilimleri Enstit s  **Kentsel Sistemler ve Ulaştırma Y netimi Anabilim Dalı**’nda **Y ksek Lisans Tezi** olarak kabul edilmiştir.

Danışman

Dr.  ğr.  yesi Ali BOYACI
İstanbul Ticaret  niversitesi

J ri  yesi

Dr.  ğr.  yesi Mustafa Cem KASAPBAşı
İstanbul Ticaret  niversitesi

J ri  yesi

Dr.  ğr.  yesi  zg r Can TURNA
İstanbul  niversitesi-Cerrahpaşı

Onay Tarihi: 24/07/2020

İstanbul Ticaret  niversitesi, Fen Bilimleri Enstit s n n 24.07.2020 tarih ve 2020/288 numaralı Y netim Kurulu Kararının 1. Maddesi gereğince, ders y klerinin ve tez y k ml l ğ n  yerine getirdiğini belirlenen “Mert Can GİRGİN” (TC: 12551079022) adlı  ğrencinin mezun olmasına oy birliğı ile karar verilmiştir.

Prof. Dr. Necip řİMřEK
Enstit  M d r 

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

24/07/2020



Mert Can GİRGIN

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vi
SİMGELER VE KISALTMALAR DİZİNİ	vii
1. GİRİŞ	1
1.1. Amaç	2
1.2. Önem	2
2. LİTERATÜR ÖZETİ	6
2.1. Akıllı Şehir Kavramı	6
2.2. Dünyadan Akıllı Şehir Örnekleri	10
2.3. Akıllı Şehirlerde Konum Temelli Mobil Uygulama Örnekleri	12
2.4. En Yakın Şehir Bileşeni İçin Uzamsal Arama Algoritmaları	14
3. UZAMSAL ARAMA ALGORİTMALARI KODLAMA ÇALIŞMALARI	16
3.1. Python Programlama Dili	16
3.2. Google Bulut Hizmetleri Sanal Makine	17
3.3. Anaconda & Jupiter Notebook	20
4. METODOLOJİ	22
4.1. k-En Yakın Komşu Algoritması	22
4.1.1. Öklid fonksiyonu ile k-en yakın komşu algoritması	24
4.1.2. Manhattan fonksiyonu ile k-en yakın komşu algoritması	25
4.1.3. Minkowski fonksiyonu ile k-en yakın komşu algoritması	26
4.2. Dörtgen Sorgular Algoritması	27
4.3. r-Ağacı Algoritması	29
4.4. kd-Ağacı Algoritması	30
5. UYGULAMA	33
5.1. Simülasyon Çalışmaları	33
5.1.1. Taksi şehir bileşeni simülasyon çalışmaları	34
5.1.2. Taksi durağı şehir bileşeni simülasyon çalışmaları	37
5.1.3. Otopark noktası şehir bileşeni simülasyon çalışmaları	40
5.1.4. Eczane şehir bileşeni simülasyon çalışmaları	42
5.2. Simülasyon Çalışmaları Bulguları	45
6. SONUÇ VE ÖNERİLER	51
KAYNAKLAR	54
ÖZGEÇMİŞ	57

ÖZET

Yüksek Lisans Tezi

DİNAMİK PARAMETRELİ UZAYSAL ARAMA İLE UYGUN DÜĞÜM SEÇİMİ ALGORİTMALARININ SİMÜLASYONU VE KARŞILAŞTIRILMASI

Mert Can GİRĞİN

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Kentsel Sistemler Ve Ulaştırma Yönetimi Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ali BOYACI

2020, 57 sayfa

Teknoloji çağı ile birlikte bilgi alış verişi büyük bir hız kazanmıştır. Hızla paylaşılan bilgiler katlanarak çoğalmış ve büyük veri kümelerini oluşturmuştur. Yaşamın her alanında ortaya çıkan büyük veri kavramı ancak teknoloji ile saklanabilir, ulaşılabilir ve incelenebilir bir hal almıştır.

Veriyi kontrol etmemizi sağlayan teknolojik gelişmeler yaşam alanlarımızda etkileyerek geliştirmiştir. Teknoloji ile bütünleşen yaşam alanlarımız “Akıllı Şehir” kavramını oluşturmuştur.

Teknoloji ile bütünleşen akıllı şehirler teknolojik gelişmeler ile paralel ilerleme göstermiştir. Artan nüfus şehir bileşeni sayısını ve yoğunluğunu arttırmıştır. Gün geçtikçe kalabalık hale gelen şehirlerde hizmete uygun şehir bileşenini bulmak büyük bir zaman kaybı haline gelmiştir. Bu sorunun çözümü için uzamsal arama algoritmalarının gücünden faydalanılmaktadır. Bu çalışmada, İstanbul şehir bileşenleri simülasyonları oluşturularak. Zaman verimliliği sağlamak için şehir bileşenlerinin yoğunluğuna göre hangi uzamsal arama algoritmasının iyi performans gösterdiği tespit etmek amaçlanmıştır.

Anahtar Kelimeler: Akıllı şehirlerde uzamsal arama, dörtgen sorgular algoritması, kd-ağacı algoritması, k en yakın komşu algoritması, r-ağacı algoritması.

ABSTRACT

M.Sc. Thesis

SIMULATION AND COMPARISON OF APPROPRIATE NODE SELECTION ALGORITHMS WITH DYNAMIC PARAMETER SPATIAL SEARCH

Mert Can GİRĞİN

**İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Urban Systems and Transportation Management**

Supervisor: Assist. Prof. Dr. Ali BOYACI

2020, 57 pages

In With the age of technology, the transmission of information has accelerated. Rapidly shared information multiplied exponentially and formed large data sets. The concept of big data emerging in all areas of life can only be stored, accessed and examined with technology.

Technological developments that enable us to control the data have improved by affecting our living spaces. Our living spaces integrated with technology have created the concept of "Smart City".

Smart cities integrated with technology have progressed in parallel with technological developments. Increasing population increased the number and density of city components. Finding the city component suitable for service has become a huge waste of time in cities that are getting crowded day by day. The power of spatial search algorithms is used to solve this problem. In this study, Istanbul city components simulations are created. It is aimed to determine which spatial search algorithm performs well according to the density of city components to ensure time efficiency.

Keywords: Kd-tree algorithm, k nearest neighbor algorithm, rectangle queries algorithm, r-tree algorithm, spatial search in smart cities.

TEŞEKKÜR

Tez çalışmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışman hocam sayın Dr. Öğr. Üyesi Ali BOYACI'ya teşekkürlerimi sunarım.

Araştırmanın yürütülmesinde manevi yardımlarını gördüğüm Gözde Merve DEMİRCİ'ye olmak üzere tüm İSBAK A.Ş. personeline teşekkür ederim.

Tezimin imalat aşamasındaki desteklerinde dolayı İstanbul Büyük Şehir Belediyesine teşekkür ederim.

Tezimin her aşamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Mert Can GİRGİN
İSTANBUL, 2020

ŞEKİLLER

	Sayfa
Şekil 1.1. Dünya çapında internet erişim oranı	4
Şekil 2.1. Akıllı şehir bileşenleri	10
Şekil 3.1. Google cloud platform	18
Şekil 3.2. Sanal makine yapılandırma	19
Şekil 3.3. Sanal makine örnekleri	20
Şekil 4.1. Yeni katılan nokta tespit koordinat gösterimi	23
Şekil 4.2. Yeni katılan noktanın koordinat sisteminde sınıflandırılması	24
Şekil 4.3. İki nokta arasındaki mesafe gösterimi	26
Şekil 4.4. P parametre değerine göre kontur eğrisi	27
Şekil 4.5. Merkez noktasına göre çizilen sanal dörtgen gösterimi	28
Şekil 4.6. r-Ağacı çalışma yapısı	30
Şekil 4.7. kd-Ağacı çalışma yapısı	31
Şekil 5.1. Taksi şehir bileşeni için r-ağacı simülasyon grafiği	36
Şekil 5.2. Taksi durağı şehir bileşeni için r-ağacı simülasyon grafiği	39
Şekil 5.3. Otopark noktası şehir bileşeni için r-ağacı simülasyon grafiği	42
Şekil 5.4. Eczane şehir bileşeni için r-ağacı simülasyon grafiği	44
Şekil 5.5. Taksi şehir bileşeni merkez nokta 1 için test süreleri	45
Şekil 5.6. Taksi şehir bileşeni merkez nokta 2 için test süreleri	46
Şekil 5.7. Taksi durağı şehir bileşeni merkez nokta 1 için test süreleri	47
Şekil 5.8. Taksi durağı şehir bileşeni merkez nokta 2 için test süreleri	47
Şekil 5.9. Otopark noktası şehir bileşeni merkez nokta 1 için test süreleri .	48
Şekil 5.10. Otopark noktası şehir bileşeni merkez nokta 2 için test süreleri.	49
Şekil 5.11. Eczane şehir bileşeni merkez nokta 1 için test süreleri	50
Şekil 5.12. Eczane şehir bileşeni merkez nokta 2 için test süreleri	50

ÇİZELGELER

	Sayfa
Çizelge 5.1. Boyutlu alan ve İstanbul şehri sınır koordinatları.....	33
Çizelge 5.2. İstanbul şehri 2B alan / şehir bileşeni oranları.	34
Çizelge 5.3. Taksi şehir bileşeni veri kümesi sınır değerleri.....	35
Çizelge 5.4. Taksi şehir bileşeni için merkez nokta değerleri	35
Çizelge 5.5. Taksi şehir bileşeni için r-ağacı koordinat değerleri.....	36
Çizelge 5.6. Taksi şehir bileşeni için kd-ağacı koordinat değerleri	37
Çizelge 5.7. Taksi durağı şehir bileşeni veri kümesi sınır değerleri.....	37
Çizelge 5.8. Taksi durağı şehir bileşeni için merkez nokta değerleri	37
Çizelge 5.9. Taksi durağı şehir bileşeni için r-ağacı koordinat değerleri.....	38
Çizelge 5.10. Taksi durağı şehir bileşeni için kd-ağacı koordinat değerleri.	40
Çizelge 5.11. Otopark noktası şehir bileşeni veri kümesi sınır değerleri	40
Çizelge 5.12. Otopark noktası şehir bileşeni için merkez nokta değerleri....	40
Çizelge 5.13. Otopark noktası şehir bileşeni için r-ağacı koordinat değerleri	41
Çizelge 5.14. Otopark noktası şehir bileşeni için kd-ağacı koordinat değerleri	42
Çizelge 5.15. Eczane şehir bileşeni veri kümesi sınır değerleri	43
Çizelge 5.16. Eczane şehir bileşeni için merkez nokta değerleri	43
Çizelge 5.17. Eczane şehir bileşeni için r-ağacı koordinat değerleri.....	44
Çizelge 5.18. Eczane şehir bileşeni için kd-ağacı koordinat değerleri	45

SİMGELER VE KISALTMALAR

dist	Mesafe
ms	Milisaniye
sn	Saniye
>	Büyüktür sembolü
<	Küçüktür sembolü
Σ	Toplam sembolü
$\wedge$	Ve koşul sembolü

1. GİRİŞ

Son yıllarda mobil teknolojilerin gelişimine bağlı olarak mobil uygulamaların hızla geliştiği ve çeşitlendiği bir dünyaya gelindi. Günümüzde geline bu noktada doğru veriye hızlı bir şekilde her yerden ulaşmak önemli hale gelmiştir. Mobil uygulamalar bireyler için hayatlarının vaz geçilmez bir parçası haline gelmiştir. Bireylerin bu tutumu sebebi ile çeşitlenen mobil uygulamalar bazı alanlarda özellikle yoğunlaşma göstermiştir. Bireylere istedikleri anda alış-veriş, iletişim, ulaşım vb. ihtiyaçları için ön plana çıkan uygulamalar bu alanda yeni bir sektör yaratmıştır. Bireylerin birçoğu dışarı çıkacakları zaman hava durumu, trafik durumu, toplu taşıma bilgisi vb. uygulamalara göz atmaktadırlar. Özellikle nüfusun kalabalık ve hayatın yoğun yaşandığı büyük şehirler ve metropoller de bireylerin bu ihtiyaçlarına cevap verebilmek adına özel sektör ve yerel yönetimler mobil uygulamalar geliştirmeye ve hizmete sunmaya başlamışlardır. Bu uygulamalar akıllı şehirler konseptinin içerisinde yer almaktadır.

İstanbul Büyük Şehir Belediyesi vatandaşlara güvenli seyahat hizmeti sunan şoför ile yolcu buluşturmaya amaçlayan iTaksi uygulamasını hizmete sunmuştur. iTaksi uygulamasıyla; yolculuk öncesi ücret hesaplama, ihtiyaca uygun taksi sınıfı seçimi, alternatif ödeme seçenekleri sunularak hızlı ve güvenli ödeme, promosyonlar sunarak hizmeti çekici hale getirme, yolculuk anında yolculuğu takip edebilme ile şoför bilgileri sunularak güvenli yolculuk yapabilme, yolculuk sonrası hizmet değerlendirme ile müşteri memnuniyeti sağlanmaktadır. Bireyler uygulama marketlerinden mobil cihazlarına indirdikleri iTaksi uygulamasına kayıt olduktan sonra taksi çağrısında bulunabilirler, uygulama bulundukları konuma en yakın taksiyi bulmak için uzamsal arama algoritmalarını kullanmaktadır. Uzamsal arama algoritmaları akıllı şehirler konsepti içerisinde yer alan tüm bileşenlerin konumlarını bulmak amacı ile kullanılabilir. Örneğin en yakın eczane, en yakın otobüs durağı, durağa gelecek olan ilk otobüsün nerede olduğu, şehirde ki vapur iskeleleri vb. bu konsept dahilinde hizmete alınacak olan tüm uygulamalar uzamsal arama algoritmaları ile hızlı ve doğru sonuca ulaşabilmektedir.

Dinamik parametrelili uzaysal arama ile uygun düğüm seçimi algoritmalarının simülasyonu ve karşılaştırılması konulu bu tezde şehir bileşenleri için oluşturulan örnek veri setleri için uzamsal arama algoritmaları uygulanacaktır elde edilen test sonuçları ile doğru algoritma seçilecektir.

1.1. Amaç

Kentsel Sistemler ve Ulaştırma Yönetimi Yüksek Lisans Programı bünyesinde hazırlanan “Dinamik parametrelili uzaysal arama ile uygun düğüm seçimi algoritmalarının simülasyonu ve karşılaştırılması” başlıklı tez çalışmasında zaman kavramının değerli olduğu büyük akıllı şehirler için hızlı bir şekilde doğru sonuç veren uzamsal arama algoritmaları modelleneyecektir. Akıllı şehir konseptinin farklı bileşenleri için yapılacak olan modellemeler ile uygun veri setleri oluşturulacaktır. Farklı uzamsal arama algoritmaları ile oluşturulan bu veri setleri üzerinde testler gerçekleştirilecektir. Test sonuçları değerlendirilerek hızlı ve doğru sonucu veren algoritmalar tespit edilecektir. Akıllı şehir bileşenlerini merkez alan veya bu konsept için geliştirilecek olan mobil uygulamalarda kullanılan uzamsal arama algoritmalarının doğru belirlenmesi ve geliştirilmesi amaçlanmaktadır.

Bu çalışma kapsamında İstanbul Büyük Şehir bünyesinde yer alan şehir bileşenleri için modellemeler yapılacaktır. Oluşturulacak olan şehir bileşeni modelleri; taksi, otobüs, durak, eczane, hastane vb. için olacaktır. Bu modeller birleştirilmiş dört farklı uzamsal arama algoritması ile teste tabi tutulacaktır. Sonuçların değerlendirilmesi ile hangi şehir bileşeni için hangi uzamsal algoritmanın daha verimli çalıştığı tespit edilerek bireyler için zamanı en verimli kullanan algoritmanın seçimi tamamlanmış olacaktır.

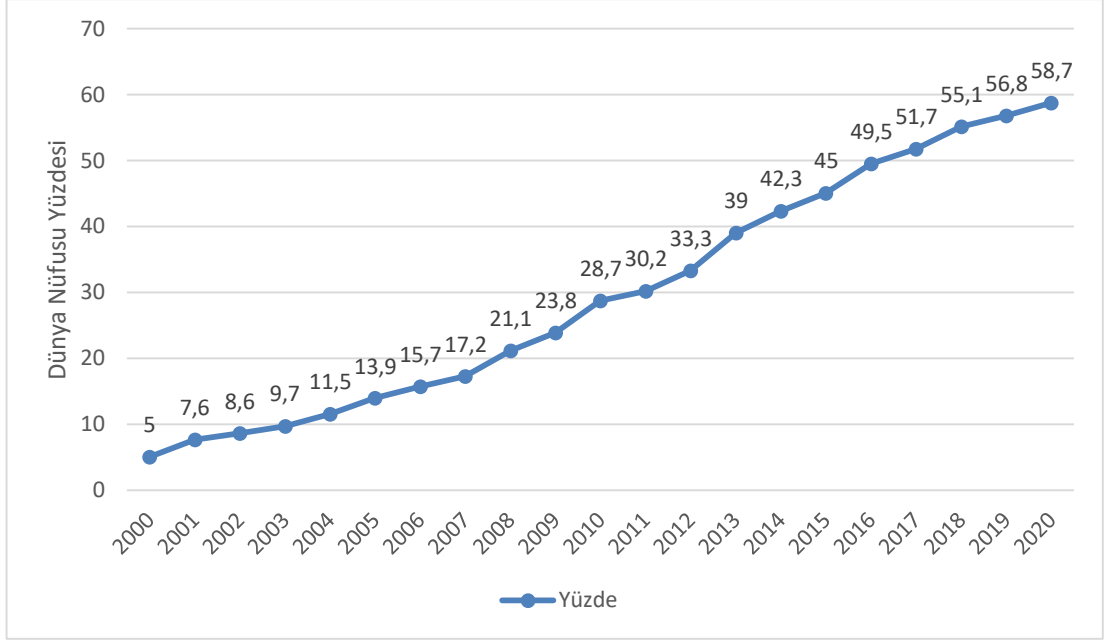
1.2. Önem

18. ve 19. Yüzyıllarda endüstri devrimi sonrası gerçekleşen teknolojik gelişmeler ışığında sanayi toplumunun temelleri atılmıştır. Sanayi toplumunda teknolojinin hızla ilerlemesi ile bilgi önem kazanmıştır. Toplum 20. yüzyılda bu çağın getirdiği

ileri teknoloji ile birlikte toplum bilgi toplumuna evrilmiştir. Günümüzde bilgi teknolojileri en karmaşık işlemde en basit işleme kadar her alanda kullanılmaktadır, çözümlenmesi gereken işlemler teknolojik cihazlar ile gerçekleştirilmektedir. Bilgiye erişimde dijitalleşen kaynaklarda önem kazanmıştır, bireyler dünyanın herhangi bir konumunda bir bilgi bankasına erişerek istedikleri bilgiye kolayca ulaşabilmektedirler (Şeker, 2005).

Endüstri devrimi sonucu kırsal kesimden kentlere başlayan göçler 20. yüzyılda ortaya çıkan teknolojik gelişmeler ile birlikte artmıştır ve kentleşmeler meydana gelmiştir (Kurt, 2006). Kentleşme ile birlikte ortaya çıkan yeni teknolojik gelişmeler olmuştur. Şehir ve teknolojiyi birleşmesi ile Akıllı Şehirler kavramı ortaya çıkmıştır. Akıllı şehirlerin odak noktası ise kentleşmenin sebep olduğu zaman kavramının verimli kullanımı zorunluluğudur. Akıllı şehirlerde zamanın verimli kullanılması bilgiye mekân fark etmeksizin hızlı ulaşabilmeye bağlıdır. Teknolojik gelişmeler ışığında gelişen internet teknolojileri, nesnelerin interneti ve donanımsal gelişmeler ile bilgiye ulaşmak için mobil cihazları en popüler araçlar haline getirmiştir.

Internet World Stats ait Şekil 1.1'de yer alan verilere göre dünya çapında internete erişim son 20 yılda ciddi bir biçimde artarak % 58,7 ye ulaşmıştır (Internet World Stats Usage and Population Statistics, 2020).



Şekil 1.1. Dünya çapında internet erişim oranı

İnternet erişiminin hızla artması ve bilgiye erişimin mobil cihazlar ile sağlanması sebebiyle mobil cihaz kullanımı da artış göstermiştir. We Are Social araştırma verilerine göre kişisel cep telefonu kullanımı 2015 yılında dünya genelinde %51 iken 2020 yılında baktığımızda dünya geneli cep telefonu kullanımı oranı % 67 ye ulaşmıştır (We are Social, 2020).

Araştırma verilerinden elde edilen istatistikler bize internet kullanımı ve buna paralel olarak cep telefonu kullanımının artış gösterdiğini göstermektedir. Şehirleşmenin hızlanması ile şehir bileşenleri de hızla gelişmekte ve çoğalmaktadır. Bireyler günlük hayatlarında şehirleşmenin getirdiği bu gelişimle internet ve mobil cihazlara daha fazla ihtiyaç duymaktadırlar. Bu süreçte akıllı şehir uygulamaları zaman tasarrufu sağlamak amacıyla daha popüler hale gelmektedir.

Özel kuruluşlar ve devlet kurumları akıllı şehir uygulamalarına daha fazla yatırım yapmaya başlamışlardır. Bu yatırımlar birçok mobil uygulamayı hayata geçirmiştir. Bunlara örnek verecek olursak; iTaksi, İBB Cep Trafik, İstanbul Eczanesi, İsbike Akıllı Bisiklet vb. uygulamalardır. Konum bilgisini aktif

kullanıldığı akıllı şehir uygulamalarında doğru ve hızlı konum hesaplaması yapmak uygulamanın performansı açısından önemlidir.

Uygulamaların performanslarını etkileyen önemli faktörlerden biri de mesafe hesaplama algoritmalarıdır. Büyük verinin yer aldığı akıllı şehir uygulamalarında performansın üst düzeyde tutulabilmesi için doğru algoritmanın seçilmesi gerekmektedir. Bu sistemler için uzamsal arama algoritmaları kullanılmaktadır ama uygulamanın kullanım modeline göre doğru uzamsal arama algoritmasının seçimi uygulamayı performans adına bir adım öne taşınmasını sağlayan bir ölçüttür. Yapılacak olan modellemeler ve bu modellemelerin üzerine gerçekleştirilecek olan testler ile hangi model için hangi uzamsal arama algoritmasının performans açısından daha verimli olduğu tespit edilecektir.

2. LİTERATÜR ÖZETİ

Bu bölümde Akıllı Şehir kavramı ve oluşumu hakkında bilgi verilecektir. Dünya Akıllı Şehirlerinde konum verisini kullanan uygulamalar örneklendirecek ve Türkiye İstanbul da kullanılan uygulamalar incelenecek. En yakın nokta hesaplamaları için uzamsal arama algoritmalarına değinilecektir.

2.1. Akıllı Şehir Kavramı

Son yüzyılda meydana gelen teknolojik gelişmeler ile kentlerde yaşam insanlar için daha cazip hale gelmeye başlamıştır. Bunun sonucu olarak kırsal alanları terk eden insanlar hayat standartlarını yükseltmek amacı ile kentlere göç etmeye başlamıştır. Günümüzde Dünya çapında nüfusun %50 den fazlası kentlerde yaşamını sürdürmektedir. 1950'li yıllarda kırsal nüfusun Dünya çapına oranı %30 bandındayken 100 yıl sonrası için 2050 yılında bu oranın % 66 bandına ulaşması beklenmektedir (World Urbanization Prospects The 2014 Revision, 2014). Şehirleşmenin getirdiği sosyal, ekonomik ve teknolojik gelişmeler insanlık için faydalı olsalar dahi beraberinde birçok sorun da getirmiştir. Şehir yönetimi birimleri sundukları hizmetlerin kontrolü ve verimliliği konusunda hızlı artan nüfus sebebi ile problemler yaşamışlardır. Bu problemleri aşmak için yönetim organları şehirleri ve hizmetleri için teknolojik çözümler geliştirmek zorunda kalmışlardır. Şehir ve teknolojinin birleşmesi ile şehirler akıllanmaya başlamıştır.

Bilgi ve iletişim teknolojilerinin gelişmesi ile akıllı kelimesi hayatlarımıza dâhil olmuş ve bu kavram ile bilgi ve iletişim teknolojilerinin harmanlandığı alanlar vurgulanmaya başlamıştır. Akıllı Şehir kavramı da bu şekilde ortaya çıkmıştır. Bilgi ve iletişim teknolojileri ile pozitif şehir odaklı teknolojik inovasyonlar şehirlerin daha rekabetçi olmalarını desteklemek anlamına da gelmektedir (Caragliu vd., 2009). Şehirlerde oluşan bu teknolojik gelişmelere örnek olarak; Akıllı Ulaşım Sistemleri, Akıllı Altyapı Sistemler, Akıllı Otomasyon ve Kontrol Sistemleri, Akıllı Kaynak Yönetimi vb. verilebilmektedir. Şehirlerin tüm bileşenlerine yayılan bu teknolojik dönüşümler ile şehirler kolay yönetilebilir, verimli ve yaşanabilir yaşam alanlarına dönüşmektedirler. Dünyada ve Avrupa

da benimsenen Akıllı Şehir kavramı; dijital şehir, sürdürülebilir şehir, bilgi şehri, eko-şehir vb. kavramlara egemen olmuştur (Manville vd., 2014). Akıllı şehirler sürdürülebilirlik hizmet devamlılığı açısından hayati önem taşımaktadır. Dijital şehir kavramının amacı ağ teknolojileri ile iş birliği, bilgi paylaşımı, tüm şehir sakinleri için birlikte devamlı çalışma ortamı sunabilmek olarak tanımlanır (Yovanof ve Hazapis, 2009). Şehirlerin kaynaklarının verimli bir şekilde kullanılarak şehir sakinlerine hizmet sunulması akıllı şehir kavramının temel amaçlarındandır. Akıllı bir şehir var olan altyapıyı yönetmek, sektörler arası iş birliğinden faydalanmak, yeni iş modelleri oluşturmayı sağlamak için bilgi ve iletişim teknolojilerinden yararlanmaktır (Kramers vd., 2014).

Akıllı şehirler yaşam kalitesini yükseltmenin yanı sıra ekonomik gelişmenin sürdürülebilirliğini de sağlayabilirler. Bu çerçevede teknolojinin gücüyle birlikte ulaşım, kamu hizmetleri, eğitim vb. şehir bileşenlerinde yüksek bir seviyeye ulaşmışlardır. Seviyenin yükselmesi için gerekli olan sosyal sermaye, kalifiye insan kaynağı, bilgi ve iletişim teknolojileri ana unsurlardır. Bu ana unsurlardan en üst düzeyde faydalanmak gerekmektedir. Literatür de birçok farklı şekilde tanımlanan “Akıllı Şehir” kavramının odak noktalarını şu şekilde özetleyebiliriz;

- “Ekonomik-politik etkinliği artırmak, sosyal, kültürel ve kentsel gelişimi sağlamak için birbirleri ile iletişim halindeki altyapıların kullanılması,
- İş odaklı yaklaşım ile (kentlerde yeni iş olanaklarının yaratılması ve geliştirilmesi) kentlerin sosyo-ekonomik gelişmişliğinin artırılması,
- Farklı kesim ve bölgelerdeki kent sakinlerinin sosyal içermeye ile kamu hizmetlerinden eşit şekilde faydalanmasının sağlanması,
- Bilgi odaklı ve küreselleşen ekonomide nitelikli insan kaynağının kentlere çekilmesi ve yaratıcı kültür ortamının oluşturulması,
- Kentsel gelişimde sosyal sermayenin önemi ve kent sakinlerinin yeni teknolojileri öğrenmesi, benimsemesi ve vatandaşların yaşayan laboratuvarlar gibi yapılar aracılığıyla kentler için yeni çözümlerin üretilmesine katkı sağlaması,
- Sosyal ve çevresel olarak sürdürülebilirliğin sağlanması, kaynakların etkin kullanılmasıdır” (T.C. Kalkınma Bakanlığı, 2013).

Akıllı şehirlerin oluşturulması dünya genelinde gerekli bir hale gelmiştir. Şehir yönetimleri ve şehre hizmet eden özel sektör teknolojik gelişmeleri yakından takip ederek bu teknolojileri şehir bileşenlerine uygulama süreçlerini başlatmakta ve uygulamaktadır. Akıllı şehirleri oluştururken sosyal medya gücü, nesnelerin interneti, büyük veri, bulut depolama ortamlar ve mobil uygulamalar vb. şehre adapte olabilecek teknolojik öğelerden yararlanılır. Akıllı Şehir kavramı kendi başına bir sektör doğurduğu için bu alanda çalışan bilim insanları, mühendisler, mimarlar ve hizmet sektörü içerisinde yer alan insanlar tarafından birçok farklı kategori akıllı şehir bileşenlerine dâhil edilmektedir. Avrupa birliği tarafından kabul görmüş olan Cohen'in "Akıllı Kentler Çarkı" yöntembilimi bir akıllı şehir için 6 temel bileşen fikrini benimser, diğer görüşlere göre daha ön planda kabul gören bu bileşenler şu şekilde sıralanır;

- Akıllı Ulaşım: Çevre dostu, bilgi ve iletişim teknolojileri destekli bütünleşmiş ulaşım ağı sistemlerini içermesi,
- Akıllı Yaşam: Bilgi ve iletişim teknolojileriyle şehir sakinlerine daha sağlıklı ve güvenilir bir ortam sağlanmasının yanı sıra insanların yaşamlarını kolaylaştırılması,
- Akıllı Yönetişim: Şehir yönetim organlarının şehir sakinleri için refahı sağlamak ve şehri çağın gerektirdiği standartlara taşıırken diğer paydaşlar ile şeffaf bir yönetim anlayışı ile hareket etmesi,
- Akıllı Çevre: Yenilenebilir enerji kaynakları, akıllı otomasyon sistemleri, akıllı çevre dostu bina ve yapılar vb. çözümlere yatırım yapılmasını kapsamaması,
- Akıllı Ekonomi: Bilgi ve iletişim teknolojileri kullanılarak üretim ve tedarik süreçlerinden yüksek verim elde etme amacı ile yapılan çalışmaları kapsamaktadır,
- Akıllı İnsanlar: Bu bileşenlerin ana odak noktası insandır, toplumun yapılan bilgi ve iletişim teknolojileri yatırımlarını kullanabilme becerilerinin geliştirilerek daha bilinçli ve kalifiye bir toplum haline gelmektir.

Literatürde bir diğer kabul gören Şekil 2.1'de gösterilen Akıllı Şehir Bileşenleri ise şu şekilde sıralanabilir; akıllı altyapı, akıllı güvenlik, akıllı enerji, akıllı

yönetim, akıllı eğitim, akıllı sağlık hizmetleri, akıllı bina ve akıllı ulaşım/hareketlilik". Bazı maddeleri özetleyecek olursak;

- Akıllı Altyapı: Şehirlerde artan nüfus ile yetersiz kalacağı ön görülen elektrik, su, gaz vb. altyapı sistemlerine sürdürülebilir, yenilikçi çözümler getirmeyi hedeflemek,
- Akıllı Şehir Güvenliği: Şehirlerde gerçekleşen beklenmedik doğal afet ve kazalara karşı şehirlerin güvenliğini, sağlığını ve sürdürülebilirliğini sağlamak amacıyla bu hassas sürecin güvenli bir şekilde yönetilmesini sağlamaktır,
- Akıllı Enerji: Günümüzde şehirlerin ihtiyaç duyduğu enerjiyi sağlamak için doğal kaynakları tüketmesi sorunu ön plandadır. Akıllı enerji çözümleri hava, su, toprak gibi doğal kaynakları koruyarak yenilikçi ve sürdürülebilir enerji çözümleri sunmayı hedefler,
- Akıllı Yönetim: Şehir yönetim organları artan nüfusun taleplerine cevap vermek için teknolojiyi iş plan ve programlarına dâhil etmektedirler. Bu çalışmaları diğer yönetim organları ile iş birliği ve şeffaflık içerisinde yapmayı hedeflemektedirler,
- Akıllı Binalar: Akıllı bina yapımı ve işletimi konularını odak almaktadır. Konutların aydınlatma, ısıtma, soğutma vb. enerji odaklı işletimleri ile giriş-çıkış denetimi yangın kontrolü, acil tahliye planlaması vb. güvenlik odaklı işletimlerini bilgi ve iletişim teknolojileri ile bütünleştirip verimli kolay yönetim sağlamayı hedeflemektedir,
- Akıllı Ulaşım: Çevre sorunlarından hava kirliliğinin önüne geçmek amacı ile sıfır karbon salınımı projelerini benimseyen akıllı ulaşım sistemleri, toplu taşımaya teşvik ederek; elektrikli ulaşım araçları, bisiklet yolları, akıllı trafik çözümleri, park et veya devam et veya toplu taşıma kullan vb. çözümler sunarak trafiği azaltarak çevreyi korumayı hedeflemektedir.



Şekil 2.1. Akıllı şehir bileşenleri

2.2. Dünyadan Akıllı Şehir Örnekleri

Londra: “Londra Çevre Stratejisi” projesi kapsamında karbon salımın oranını sıfıra indirmeyi ve %50 yeşil alan yaratmayı hedeflemektedirler. Şehir de elektrikli araç kullanımı teşvik edilerek karbon emisyonun sıfır olduğu yeşil bölgeler yaratıp sadece elektrikli araçların girebileceği bölgeler oluşturulmaktadır. Akıllı şehir kapsamında destek sağlaması amacıyla “Açık Veri Enstitüsü” kurulmuştur, kar amacı gütmeyen bir kuruluştur. Güvenlik şeffaflığı adına güvenlik güçlerinin vücutlarına takılan güvenlik kameraları hizmete alınmıştır. Zaman tasarrufu amacı ile toplu taşımada temassız ödeme seçeneği sunan Oyster Kart hizmete sunulmuştur. Engelli veya Dezavantajlı bireyler için Londra metrosuna navigasyon sistemi kurulmuştur. Şehrin planlanmasına şehir sakinlerinin de dâhil edilmesini amaçlayan “Londra ile Konuş” soru-cevap anketleri yapılmaya başlanmıştır. “London Data Store” ve “Croydon Teknoloji Merkezi” kurularak büyük verinin verimli incelenmesi ve yenilikçi çalışmaları hızlı şekilde gerçekleştirip şehrin ihtiyaçlarına cevap verilmesi amaçlanmıştır.

Barselona: Enerji anlamında kendi kendine yetebilen bir şehir olarak göze çarpmaktadır. TERSA Atıklardan Biogaz Enerji Üretme Tesisi şehrin enerji

ihtiyacına cevap vermektedir. FapLab aracılığı ile kurulan ağlar sayesinde akıllı üretim teknolojilerinin kullanımı ve etkileri detaylı olarak gözlenmektedir. Akıllı su yönetim sistemi Sentilo sayesinde kaynaklar verimli kullanılabilir. Halka açık internet hizmeti sağlanmaktadır. Akıllı şehir fuarları ve mobil iletişim için küresel sistemler kongresi gibi akıllı şehirler etkinlikleri ile öncü bir şehirdir.

Paris: Akıllı ulaşım konusunda teknolojiler geliştirmeyi hedefleyen şehir; 2007 yılında ket çapında bisiklet paylaşım programı başlatmıştır. IssyGrid Projesi kapsamında akıllı şebeke araştırma ve geliştirme projesi tasarlanmış ve Issy-les-Moulineaux bölgesinde faaliyete alınmıştır (Issycom, 2007).

Seul: Bilgi ağı köyü projesi kapsamında kırsal bölgelerinde akıllı tasarlanması hedeflenmektedir. Elektrikli araçlar teşvik edilerek, yolları elektrikli araçları şarj edebilir şekilde dizayn edilmiştir. Bu şekilde sıfır karbon salınımı ve hava kirliliğini engellemek hedeflenmektedir. Şehirde ücretsiz kablosuz internet hizmeti ve açık veri hizmeti sunularak şehir sakinlerinin büyük veriye hızlı ulaşımını amaçlanmıştır.

Viyana: 2050 Viyana Akıllı Şehir hedefi kapsamında 100 aktif proje üzerinde çalışmaktadır. Elektrikli araç kullanımını teşvik etmektedir ve şehrin birçok konumunda elektrikli araç şarj istasyonları yer almaktadır. Bisiklet paylaşım programının yanı sıra araç paylaşım programını da şehir içerisinde teşvik etmekte ve bu programı şehre gelen ziyaretçilerinde hizmetine sunmaktadır.

Singapur: Şehir de yer alan konut ve yapılar büyük ölçüde çevreye uyumlu akıllı bina konseptindedir. Şehir yönetim hizmetlerin %98 i çevrim içi olarak şehir sakinlerine sunulmaktadır. Şehrin her yanına dağılmış olan fiber optik kablolar ile şehir hakkında gerçek zamanlı bilgi toplanmaktadır. Akıllı ulaşım kapsamında teknolojik yolları gelişmiş bir ulaşım ağına sahiptir.

Toronto: Şehir yönetim organları, kamu yönetim birimleri ve özel sektörün iş birliği ile karbon salınımının sıfıra indirilmesi için teşvik çalışmaları yürütülmektedir. Şehirde atıklardan elde edilen gazlardan enerji sağlanması ön

plana çıkmaktadır. Atık toplama araçlarının yakıtları da bu atıklardan elde edilen gazlardan sağlanmaktadır.

Tokyo: Japonya da kırsal kesimin gelişmesi adına banliyö bölgelerine yönelik akıllı şehir sistemleri tasarlamaktadır. Şehir sakinlerinin kullanımı için birçok mobil uygulama geliştirmiştir ve geliştirmeye devam etmektedir. Şehir yönetim organları özel sektörle iş birliği içerisinde güneş enerjisi kullanımı ve depolama teknolojilerini kapsayan teşvikler yapmaktadır.

Akıllı şehir hedeflerini gerçekleştirmek için enerji, çevre, ulaşım vb. konularda çalışmalarını sürdüren Kopenhag, Stockholm, Amsterdam ve Helsinki de birçok tasarım yapıp hayata geçirmiştir. Akıllı şehirler kapsamında birçok şehir yenilikçi çalışmalarını sürdürmektedir.

2.3. Akıllı Şehirlerde Konum Temelli Mobil Uygulama Örnekleri

Dünya genelinde mobil uygulamaların akıllı şehirler ile harmanlamasından sonra şehir yönetim organları ve özel sektör tarafından birçok akıllı şehir mobil uygulaması hizmete sunuldu. Şehir sakinleri tarafından ilgi gören uygulamaların başını taksi çağırma uygulamaları çekmektedir. Taksi çağırma, akıllı ulaşım ve konum temelli uygulamaları şu şekilde inceleyebiliriz;

- Uber: Uber Teknoloji firması bünyesinde dünyanın birçok ülkesinde insan taşımacılığı amacı ile araç ve şoför hizmeti bulmayı sağlayan bir mobil uygulamadır. Bir tuşa dokunarak araç çağırma kolaylığı sağlamak vizyonu ile hizmete sunulan uygulama dünya çapında gördüğü ilgi sayesinde dünya çapında etkileşim sağlamıştır (Uber, 2009). Güvenlik, şeffaflık, değerlendirme ve kolaylık odaklı hizmet anlayışı sunmaktadır.
- Lyft: Lyft şirketi bünyesinde hizmete sunulan uygulama Uber den sonra Amerika'nın 2. Büyük araç çağrı şirketidir. Araç çağrı hizmeti yanı sıra akıllı ulaşım kapsamında yer alan bisiklet paylaşım hizmeti de sunmaktadır. Mobil uygulamadan yemek dağıtım hizmeti de alınabilmektedir. Hızlı, verimli ve güvenli ulaşımın yanı sıra güvenli

ödeme, kalifiye çalışan ve sürekli genişleyen hizmet alanı sunmaktadır (Lyft, 2020).

- Line Taxi: Asya pazarının popüler mesajlaşma uygulaması LINE akıllı şehirler yatırım yapan bir başka şirkettir. LINE Taxi mobil uygulaması ile akıllı ulaşım sektörüne adım atmış oldu. Şirket bünyesinde yer alan ödeme sistemi LINE Pay ile entegre olarak çalışabilen sistem ödeme kolaylığı ve güvenliği sunmaktadır (Line, 2020).
- Ola: Hindistan pazarına büyük pay sahibi olan taksi çağırma uygulamasıdır. Acil durum düğmesi, sürüş takibi, ödeme gizliliği ve kalifiye personel ile güvenli hizmet sunmayı hedeflemektedir. Şirketler için çalışan taşımacılığı anlamında kurumsal destek hizmeti sunarak şirketler için ekonomik hizmet sağlamayı amaçlamaktadır (Ola,2020).
- iTaksi: İstanbul Büyük Şehir Belediyesinin bir hizmeti olarak Türkiye, İstanbul konumunda faaliyet gösteren uygulama şehir yönetim organları tarafından faaliyete geçirilen ve yönetilen bir uygulamadır. Alternatif ödeme seçenekleri, promosyon imkanları, güvenli yolculuk, yolculuk hesaplama, yolculuk takibi ve ihtiyaca uygun araç sınıfı seçimi sunarak şehirde yer alan Büyük Şehir Belediyesi bünyesinde ki taksileri tek bir uygulamada toplamıştır. Güvelik odaklı hizmet sunan uygulama kamera kayıt, acil durum düğmesi ve ana merkezden yolculuk takibi gibi özellikleriyle ön plana çıkmaktadır (iTaksi,2020).
- İstanbul Eczanesi: İstanbul Büyük Şehir Belediyesinin bir başka hizmeti olan bu uygulama akıllı sağlık hizmetleri yanı sıra ulaşım kolaylığı da sunmaktadır. Bu uygulama ile 2-10 km arası mesafede ki yakın eczaneler bulunabilir. Nöbetçi eczane veya en yakın eczane bulunabilir, bu eczaneler hakkında bilgi alınabilir veya ulaşmak için en yakın rota oluşturulabilir (İBB Eczane Mobil Uygulaması, 2020).
- İspark: İstanbul Büyük Şehir Belediyesinin iştiraki İstanbul Otopark İşletmeleri tarafından geliştirilip hizmete sunulan bu uygulama ile İstanbul'daki otoparkların yoğunluk durumları, konumumuza yakın olan otoparklar ile otoparklara nasıl ulaşacağımız ile ilgili rota bilgisine erişebilmekteyiz.

- İsbike: İstanbul Büyük Şehir Belediyesi tarafından hizmete sunulan bir başka akıllı şehir ulaşım hizmeti uygulamasıdır. Uygulama ile en yakın istasyon ve istasyonda ki bisiklet sayısına ulaşılabilir. Geçmiş kullanım verilerine de erişme olanağı sunulmaktadır (“İsbike Nasıl Çalışır?”, 2020).

2.4. En Yakın Şehir Bileşeni İçin Uzamsal Arama Algoritmaları

Akıllı Şehirler kapsamında yer alan akıllı ulaşım hizmetleri mobil uygulamalar ile şehir sakinlerinin kullanımına sunulmaktadır. Bu uygulamaların örneklerini incelediğimizde temel amaçlarının zaman verimliliği sağlamak olduğunu görmekteyiz ancak bunun ardında akıllı şehir konseptinin getirdiği çevre ve enerji verimliliği gibi hassas konularda yer almaktadır. Minimum enerji ile maksimum verimi hızlı bir şekilde sağlamak için konumsal hesaplamalar yapan bu uygulamalar uzamsal arama algoritmalarından güç almaktadırlar. Uzamsal arama algoritmasının performansı uygulamanın performansını belirlemektedir. Sunulan hizmet için doğru uzamsal arama algoritmasının seçimi önemli bir faktördür.

Uzamsal arama algoritmaları büyük veri ile birleştğinde donanımsal olarak yüksek işlemci gücüne ihtiyaç duysa da optimum uzamsal arama algoritmasının seçimi donanımsal geliştirmeden daha önemli bir husustur. En yakın komşu hesaplayan algoritmaların bazıları basit hesaplama yöntemleri ile hızlı sonuç verseler dahi her veri modeli için en iyi çözüm değildir.

Veri madenciliği ve istatistiksel hesaplamalarında kullanılan popüler yöntemlerin başında “k En Yakın Komşu” algoritmaları gelmektedir. k en yakın komşu algoritmaları olağan üstü sınıflandırma ve hesaplama performansları sebebi ile tercih edilirler. k en yakın komşu hesaplamalarında kullanılan k değeri uzmanlar tarafından belirlenseler dahi tüm test modelleri için en iyi çözüm sunan algoritma olmayabilir. Sabit bir k değeri ile bu algoritmaların uygulanması yararlı değildir (Zhang vd., 2017).

Literatürde yer alan çalışmalar incelendiğinde k en yakın komşu algoritmaları dörtgen sorgular birleştirildiğinde işlem ve zaman maliyetlerinin düşmesi beklenmektedir. Temel arama algoritmalarının belirlediği strateji ağaç budama tekniği uygulayarak performans artışı sağlamaktır. Büyük veride nesnelerin sıklığı harcanan zamanı arttırmaktadır. Sık ve büyük veride ızgara sistemi ile arama alanında bölümlendirme yapıp k en yakın komşu algoritması ile birleştirildiğinde bazı ızgara bölümleri boş veya boşa yakın kalacağı için düşük depolamaya ve esnek olmayan yavaş aramaya sorgusuna sebep olacaktır (Li ve Tang, 2010). Bu sebepten dolayı k en yakın komşu algoritması dörtgen sorgular ile güçlendirilmiş, veri modelinin istatistiksel nokta dağılımına göre genişletme aralığı belirlenip “ k en yakın komşu ile dörtgen sorgular” kullanımı fikri ortaya çıkmıştır (Liu vd., 2002).

Başka bir çalışmada ise büyük boyutlu alanlarda yapılacak olan uzamsal aramalarda zaman maliyetini etkili bir biçimde azaltmak için r -Ağacı algoritmalarına başvurulmuştur. r -Ağacı algoritmalarının testleri incelendiğinde daha yüksek bir mekânsal alan kullanımı desteklediği ve zaman maliyetlerini azalttığı sonucuna ulaşılmıştır. Aynı zamanda arama sorgularının sayısının arttırılabileceği gözlemlenmiştir. Bu nedenle r -ağacı algoritmaları etkili ve doğru bir yöntem olarak önerilmektedir (Zhang vd., 2007). Geniş alanlarda arama yapılması gereken durumlarda r -ağacı ile oluşturulacak olan düğümlerden boş olanlarda hiçbir zaman arama yapılmayacağı için bu algoritmalar büyük mekânsal sorgular için umut verici yöntemlerden biridir (Zhu vd., 2007).

Literatürde tercih edilen bir diğer uzamsal arama algoritması ise kd -ağacı algoritmasıdır. Yapılan testlerde kd -ağacı algoritması ile k en yakın komşu algoritmaları birleştirilerek uygulanmaktadır. Büyük veri modelleri için benzersiz bir depolama yöntemi sunmaktadır. Ağaç algoritmalarının ve k en yakın komşu algoritmalarının güçlü yönlerinin birleştirildiği yeni bir uzamsal arama algoritması oluşturulmaktadır. Bu yeni algoritma “ kd -tree- k en yakın komşu” algoritması olarak adlandırılır (Hou vd., 2019).

3. UZAMSAL ARAMA ALGORİTMALARI KODLAMA ÇALIŞMALARI

Bu bölümde tez çalışmasında kullanılan uzamsal arama algoritmalarının kodlanması için kullanılan teknolojiler ve platformlar yer almaktadır.

3.1. Python Programlama Dili

Bilgisayarın hayatımıza dâhil olması ile birlikte bu teknolojiyi bir adım öne taşımak için birçok çalışma yapılmıştır. Programlama dillerinin geliştirilmesi ve yeni programlama dillerinin yazılması da bu gelişimler arasındadır. Guido Van Rossum 1990'lı yılların başlarında Python programlama dilini yaratmak için geliştirmeler yapmaya başlamıştır. Python programlama dilini doğuşunda genel kullanım amacına hizmet eden ABC programlama dilinin etkisi büyüktür (Dierbach, 2012). Genel hizmet amacı ile kullanılan ABC dilinin geliştirme açısından elverişsiz olduğu kanısına varan Rossum ABC dilinin güçlü yanlarını alarak temel amacı daha okunaklı kod yazmak olan Python dilini geliştirmiştir.

Temel hedefi okunaklı kod yazmak olan Python dili sade bir söz dizimi ile nesne tabanlı dinamik bir dil olarak karşımıza çıkmaktadır. Düzenli kod yazma alışkanlığı kazandırmak amacı ile Python dilinde girinti hiyerarşisi önem arz etmektedir. Satırlar arasında düzenli verilen girintiler kodun okunabilirliğini arttırmaktadır. Nesne tabanlı bir dil olan python da değişken tanımlamak için farklı veri tipleri kullanılmaktadır. Veri tipleri birer sınıfı temsil ettiğinden dolayı tanımlanan değişkenlerde bir sınıfa ait olan nesneleri oluşturmaktadır. Nesneye dayalı birçok dile kıyasla python dili dinamik yapısı sebebi ile değişkenlerin veri tiplerini programın başında tanımlama ihtiyacı duymaz. Değişken tip kontrolleri derleme anında değil programın çalıştığı anda yapılmaktadır (Tratt, 2009).

Python kendi içeriğinde bulunun birçok paket program aracılığı ve açık kaynak kodlu bir programlama dili olduğu için birçok serbest veya kurumsal geliştirici tarafından geliştirilen zengin modülleri sayesinde geniş bir yelpazede çalışma imkânı sunmaktadır. Örneğin algoritma geliştirme çalışmalarında matematiksel

işlemler için kullanılan modülü kodumuza dâhil etmek için “import math” yazmak yeterlidir.

Bilim insanları ve mühendisleri üzerlerinde çalıştıkları veri modelleri için birçok farklı platform denemişlerdir. Okunurluk, dinamiklik ve söz dizimi gibi öne çıkan özellikleri ile python özellikle büyü veri modelleri üzerinde çalışılırken zaman kaybını en aza indiren bir performans sergilemektedir. Bu benzersiz özellikleri Python programlama dilini bilim insanları ve mühendisler için iyi çalışma platformu sunan bir seçenek haline getirmektedir (Oliphant, 2007).

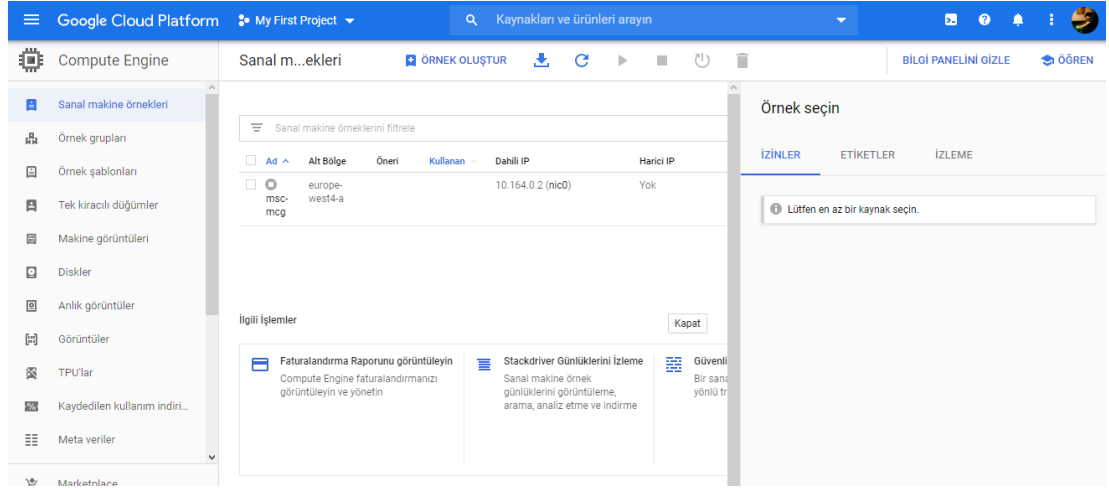
3.2. Google Bulut Hizmetleri Sanal Makine

Son yıllarda internet tabanlı sektörlerin artması ile birlikte bilgi ve iletişim teknolojileri şirketlerinde büyümeler yaşanmıştır. Çevirim içi ortamlarda yapılan hareketler sanal olarak gözükseler dahi aslında bu işlemlere imkân tanıyan platformlar fiziksel olarak var olmak zorundadırlar. Şirketlerin fiziksel olarak bu platformları sahip olmaları maliyet açısından negatif etki yaratmaktadır. Maliyetleri düşürmek için farklı çözümler sunan Google vb. firmalar bulut hizmetleri üzerinde araştırma ve geliştirmeler yaparak bu alanda hizmetler sunmaktadırlar.

Google şirketinin sunduğu bulut hizmetlerinin biride sanal makine hizmetidir. İhtiyaç doğrultusunda ki sanal makinayı bulut hizmeti üzerinden kiralama imkânı sunmaktadır. Kullanım amacına göre işletim sistemi seçeneği, sisteme bağlanma seçenekleri, sanal makinanın işlemci, depolama vb. donanımsal kapasitelerinde esneklik gibi özellikleri sunarak istenildiği takdir de bu özellikler üzerinde değişiklikleri hızlı bir şekilde yapabilmeyi sağlamaktadır.

İhtiyacımız doğrultusunda bir sanal makine oluşturmak için Google Bulut Platformuna giriş yapmamıza olanak sağlayan yetkili bir yönetici hesabına ihtiyacımız vardır. Bunu Google’ın elektronik posta hizmeti olan gmail ortamından sağlayabiliriz. Yetkili hesap bilgileri ile platforma giriş yaptıktan sonra konsol sayfasına ulaşılır. Şekil 3.1’de gösterilen konsol sayfasında

“Compute Engine” bölümü altından “Sanal Makine Örnekleri” sayfasından yeni sanal makine oluşturma imkânı sağlanmaktadır.



Şekil 3.1. Google cloud platform

Sıradaki işlem yeni sanal makine için yapılandırmadır. Bu bölümden sanal makine adı, makinenin yer alacağı fiziksel konum için bölge ve alt bölge, işlemci, depolama, bellek, ön yükleme diski, erişim türleri ve güvenlik ayarları yapılabilmektedir.

Ad [?]
Ad değiştirilemez

Etiketler [?] (isteğe bağlı)

+ Etiket ekle

Bölge [?]
Bölge değiştirilemez

Alt bölge [?]
Alt bölge kalıcıdır

Makine yapılandırması

Makine ailesi

Genel amaçlı
Bellek açısından optimize edilmiş
İşlem açısından optimize edilmiş

Maliyet ve esneklik için optimize edilmiş, yaygın iş yüklerine yönelik makine türleri

Diziler

Intel Skylake CPU platformu veya daha eski sürümlerinden biri tarafından desteklenmektedir

Makine türü



vCPU
1

Bellek
3,75 GB

CPU platformu ve GPU

Kapsayıcı [?]
☐ Bu sanal makine örneğine bir container dağıtın. [Daha fazla bilgi](#)

Önyükleme disk [?]



Yeni 10 GB standart kalıcı disk
Görüntü
Debian GNU/Linux 9 (stretch)

Değiştir

Kimlik ve API erişimi [?]

Hizmet hesabı [?]

Erişim kapsamı [?]

☒ Varsayılan erişime izin ver
☐ Tüm Cloud API'lerine tam erişime izin ver
☐ Her API için erişimi ayarla

Güvenlik Duvarı [?]
İnternet'ten belirli ağ trafiğine izin vermek için etiketler ve güvenlik duvarı kuralları ekleyin

☐ HTTP trafiğine izin ver
☐ HTTPS trafiğine izin ver

Yönetim, güvenlik, diskler, ağ iletişimi, tek kiracılı

Bu örnek için faturalandırılacaksınız. [Compute Engine fiyatlandırması](#) [?]

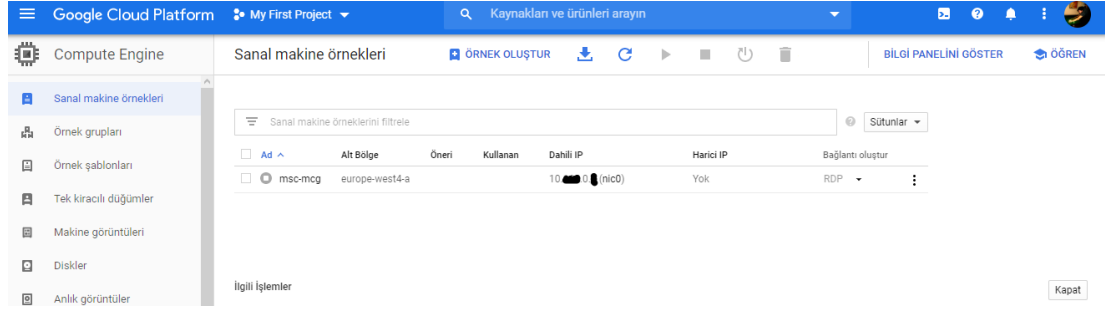
Oluştur

İptal

Şekil 3.2. Sanal makine yapılandırma

19

Sanal makine istenilen özellikleri ile oluşturulduktan sonra Şekil 3.3’de gösterilen “Sanal Makine Örnekleri” bölümünden makineye ait özet bilgilerine ulaşılabilir. IP adresi ve sistem tarafından verilen erişim şifresi ile kullanıcı ara yüzü olan sistemlere uzak masa üstü bağlantısı aracılığı ile bağlanılabilmektedir.



Şekil 3.3. Sanal makine örnekleri

3.3. Anaconda & Jupiter Notebook

Python dilinde programlama yapmak isteyenler için açık kaynak kodlu paketler sunan platformlar mevcuttur veri bilimi, veri analizi vb. bilimsel uygulamalar üzerinde çalışmak isteyenler için ortam yönetimi sağlayan Anaconda tümleşik bir platform olarak iyi bir seçenektir.

Anaconda platformunun avantajlarından biride sistem için uygun sürüm seçilip kurulup işlemi tamamlandığında python kurulumu yanı sıra jupyter, spyder vb. araçlarında kurulmuş olarak kullanıma hazır hale gelmesidir. Platformun avantajlarına devam edecek olursak; yüksek işlem performansı sağlar, paketler arası uyumlulukları kontrol eder olası uyumsuzluklar hakkında bilgi vererek bunları giderebilir, iPython sürümlerini destekleyen IBM, Google Colab, Microsoft Azure vb. ortamlara sorunsuz ulaşım ve aktarım sağlamaktadır.

Platform ile birlikte kurulan araçlardan Jupyter Notebook web tarayıcı üzerinden belge formatında ki dosyaları ve kodları düzenlemeye yarayan sunucu-istemci uygulamasıdır. Python dilini destekleyerek hizmete giren araç destek alanını genişleterek Haskel, Ruby, R vb. veri bilimi için kullanılabilen popüler dillere de hizmet vermeye başlamıştır (Doukas, 2020). Adından da yer alan “Notebook”

kelimesi not defteri anlamına gelmekte ve bir not defterine yazar gibi kod yazma imkânı tanımaktadır. Yazılan kodlar parçalar halinde bölümlendirilebilir ve bölüm bölümde çalıştırılabilmektedir. Kodların çıktılarını istenilen formata dönüştürülerek görüntüleme imkânı tanır. Bu özellikleri veri bilimi üzerine çalışan bilim insanları ve mühendisler tarafından tercih edilmesine sebep olmaktadır.

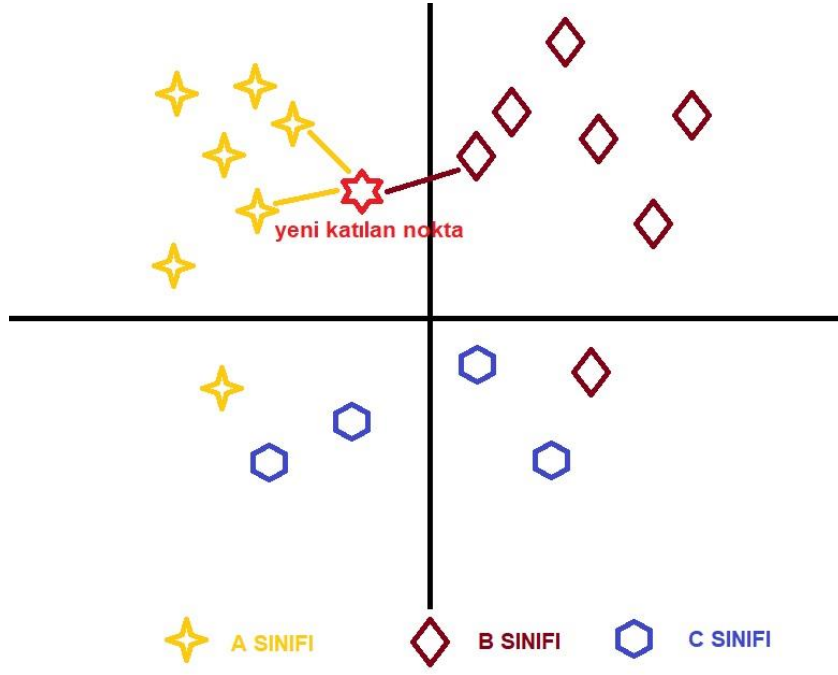
4. METODOLOJİ

Bu bölümde uygulama öncesi belirlenen ve geliştirilen yöntembilimleri anlatılmaktadır.

4.1. k-En Yakın Komşu Algoritması

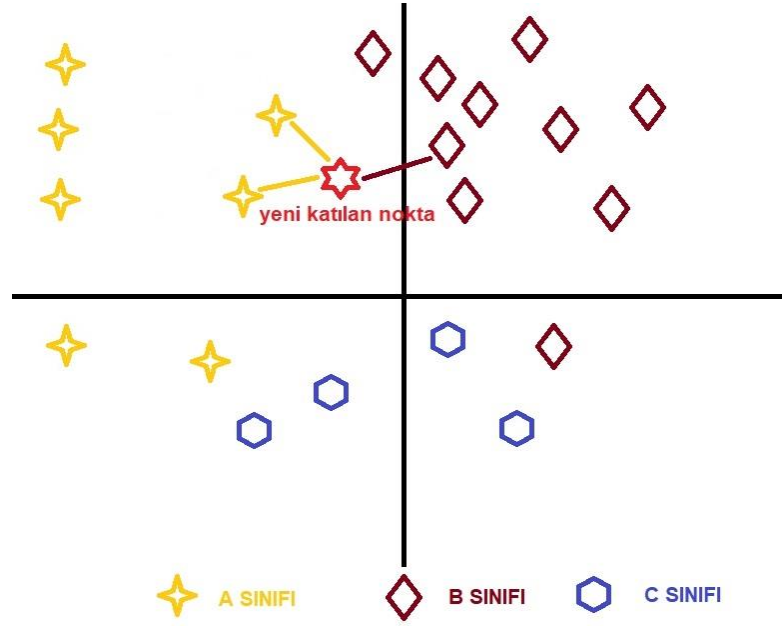
k-en yakın komşu algoritması kolay uygulanabilen öğrenme temelli bir algoritmadır. k-en yakın komşu algoritmaları geleneksel parametrik olmayan sınıflandırma yöntemi algoritmaları olarak kullanılmaktadır (Cover ve Hart, 1967). Sınıflandırma işlemlerinin yanı sıra regresyon problemlerinin çözümünde de aktif rol almaktadırlar. Veri bilimi amaçlı çalışmalarda sınıflandırma problemlerinin çözümünde kullanılmasına daha sık rastlanmaktadır.

k-en yakın komşu algoritmaları sınıflandırma işlemlerini yapmak için sınıfları belli olan bir örnek veri kümesinden faydalanmaktadırlar. Kullanılan veri kümesine hesaplamalar için yeni bir veri katılması gerekmektedir. Yeni katılan veri merkez nokta ve veri kümesindeki verilerde nokta olarak kabul edilir. Merkez noktanın diğer noktalara olan uzaklığı hesaplanır. Algoritmanın adında yer alan “k” değeri sınıflandırma işlemi yaparken kaç en yakında yer alan kaç adet noktanın sınıfına bakılacağını temsil eder. Örneğin Şekil 4.1’de k değerini 3 alarak yeni katılan noktanın hangi sınıfa ait olduğu belirlenmiştir.



Şekil 4.1. Yeni katılan nokta tespit koordinat gösterimi

Yeni katılan noktaya en yakın 3 noktaya baktığımızda 2 adet A sınıfı elemanına ve 1 adet B sınıfı elemanına yakındır. Bu sebepten dolayı A sınıfı elemanı olarak sınıflandırılır. Ancak Şekil 4.2’de örnek incelendiğinde k değeri 3 alındığında yeni katılan nokta A sınıfı elemanı olarak sonuçlansa bile bu doğru bir sınıflandırma olmayabilir. Sınıfları oluşturan noktaların geneline bakıldığında yeni katılan noktanın aslında B sınıfı elemanlarının yoğunluklu olduğu bölgeye daha yakın şekilde konumlandığı ve bu sınıfa ait olduğu iddia edilebilir. Bu durum gösterir ki veri kümesi için k değerinin seçimi de doğru sınıflandırma için büyük önem arz etmektedir.



Şekil 4.2. Yeni katılan noktanın koordinat sisteminde sınıflandırılması

Sınıflandırma işlemlerini yapabilmek için k-en yakın komşu algoritmasının içerisinde uzaklık hesaplama fonksiyonları kullanılır. Merkez nokta ve diğer noktalar arası uzaklık hesaplaması için genelde Öklid fonksiyonu kullanılmaktadır. Bu algoritmanın önemli noktalarından biri olan performanslı çalışma ölçütünü üst seviyede tutmak için Öklid fonksiyonu dışında farklı uzaklık hesaplama fonksiyonları da kullanılmaktadır. Çalışmanın uygulama bölümünde k-en yakın komşu algoritması içerisinde 3 farklı uzaklık hesaplama fonksiyonu kullanılacaktır.

- Öklid Uzaklık Fonksiyonu
- Manhattan Uzaklık Fonksiyonu
- Minkowski Uzaklık Fonksiyonu

4.1.1. Öklid fonksiyonu ile k-en yakın komşu algoritması

Uzamsal arama algoritmalarında yer alan mesafe hesaplamaları için Öklid fonksiyonu kullanılabilir. Algoritma içerisinde Öklid fonksiyonu kullanıldığı zaman iki nokta arasında ki doğrusal uzaklık hesaplanmaktadır. A merkez nokta ve diğer nokta B olarak temsil edilirse Öklid fonksiyonunun genel kullanımı şu şekildedir; (Hu vd., 2016).

$$dist(A, B) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (4.1)$$

Burada, x ve y koordinatları, i değişkeni, m veri kümesindeki nokta adedini ifade etmektedir.

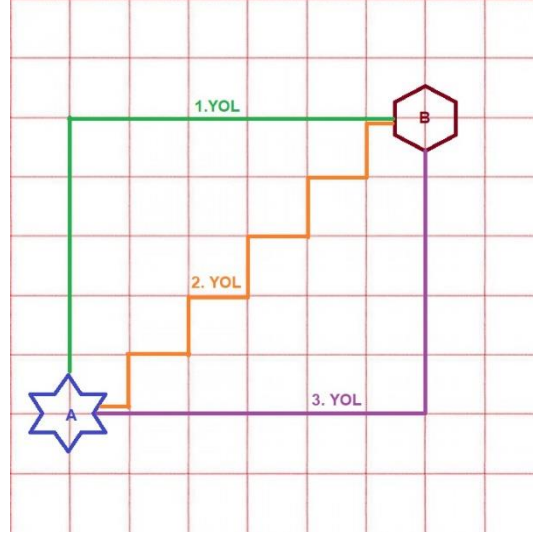
4.1.2. Manhattan fonksiyonu ile k-en yakın komşu algoritması

Manhattan uzaklık fonksiyonu Öklid fonksiyonu yerine alternatif olabilecek bir başka fonksiyondur. “Taxicab Geometrisi” olarak da bilinen bu yöntemin temelinde 2 nokta arasında ki yatay ve dikey uzunlukların toplamını hesaplamak vardır. Matematiksel olarak formüle dökülmüş hali şu şekildedir;

$$dist(A, B) = |x_A - x_B| + |y_A - y_B| \quad (4.2)$$

Burada, a ve b değerleri koordinat sistemindeki 2 farklı noktayı temsil etmektedir.

Bir şehrin yolları ızgara benzeri bir yapı şeklinde tasarlandığında 2 nokta arası mesafe için takip edilecek yolun uzunluğunu temsil etmektedir. Öklid fonksiyonu gibi iki nokta koordinatları arasındaki uzaklıklara çapraz bir çizgi çizip hipotenüs oluşturamaz. Bu sebepten dolayı iki nokta arasında ki doğrusal en kısa mesafeyi vermek yerine koordinatları arasındaki uzaklıkların toplamını verir. k-en yakın komşu hesaplamalarında tüm veri kümesi için uygulandığı zaman en yakın noktayı hesaplama konusunda sorun yaşanmaz. Şekil 4.3’de görüldüğü gibi A noktasından B noktasına 3 farklı yol tercih edilse dahi her bir yolun yatay ve dikey ilerleme uzunlukları toplandığında aynı sonuç çıkmaktadır.



Şekil 4.3. İki nokta arasındaki mesafe gösterimi

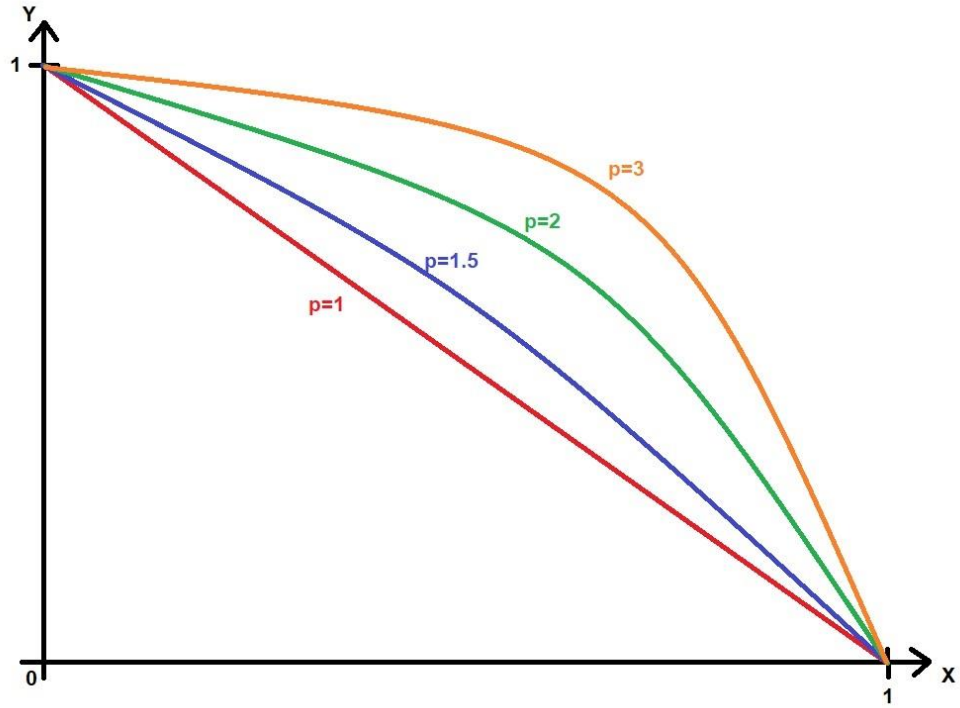
4.1.3. Minkowski fonksiyonu ile k-en yakın komşu algoritması

Öklid ve Manhattan fonksiyonlarının harmanlanmış hali olarak gözüken bir diğer uzaklık hesaplama fonksiyonu ise Minkowski fonksiyonudur. Matematiksel formülünde yer alan parametrik artan “p” değerine bağlı hesaplama yapmaktadır;

$$dist(A, B) = (\sum_{i=1}^n |x_i - y_i|^p)^{1/p} \quad (4.3)$$

Burada n veri kümesindeki nokta adedini, p parametre değişkenini temsil etmektedir.

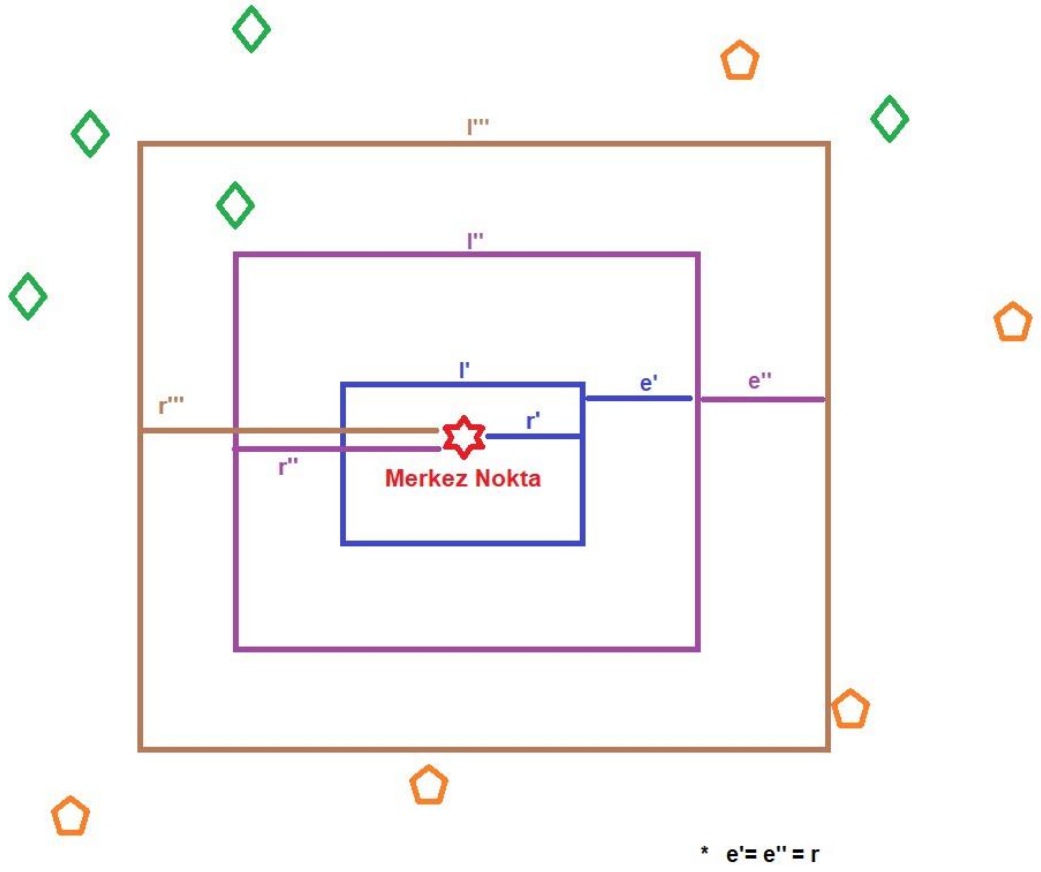
Öklid ve Manhattan ile bağdaştırılmasının sebebi ise; Formülünde yer alan “p” değerini 1 alırsak fonksiyon Manhattan fonksiyonu, “p” değerini 2 alırsak fonksiyon Öklid fonksiyonu olarak çalışıp işlem yapmaktadır. Şekilde görüldüğü gibi “p” değerlerine göre yapılan hesaplamalarda oluşan kontur eğrisinde farklılıklar oluşmaktadır. Minkowski uzaklığının genel formülü olsa dahi “p” değerinde ki bu değişimler ile çeşitli uzaklık ölçütleri ifade edilebilmektedir (Taşcı ve Onan, 2017).



Şekil 4.4. P parametre değerine göre kontur eğrisi

4.2. Dörtgen Sorgular Algoritması

Merkez nokta etrafına çizilen ve parametrik olarak genişletme yapılarak uygulanan sorgulardır. En yakın komşuyu aramak için merkez nokta etrafına bir sanal dörtgen çizilir. Bu dörtgen içerisinde nokta olup olmadığı kontrol edilir eğer nokta var ise sadece alan içerisinde ki noktalar için mesafe hesaplama fonksiyonu uygulanır. Merkez nokta etrafına çizilen dörtgenin kapsadığı alan içerisinde herhangi bir komşu nokta yer almaması durumunda ise merkez nokta belirlenen çarpan katsayısı ile çarpılarak genişletilir. Her genişletilme işlemi sonrası tekrar alan içerisinde komşu nokta olup olmadığı kontrol edilir, komşu nokta bulunduğu takdirde mesafe hesaplama fonksiyonları uygulanır. Bu yaratılan döngü alan içerisinde bir komşu nokta tespit edilene kadar veya istenilen adette döngü sayısına ulaşılan kadar devam etmektedir. Şekil 4.5'te gördüğümüz gibi her bir döngü aşaması dörtgeni genişletmekte ve dörtgen sınır mesafesini genişletme katsayısı kadar arttırmaktadır.



Şekil 4.5. Merkez noktasına göre çizilen sanal dörtgen gösterimi

Sanal dörtgen çizme ve genişletme işlemleri için gerekli olan formülde döngü sayısı “I”, genişleme katsayısı “e”, dörtgen sınır mesafesi “r” aşağıdaki gibi tanımlanır;

$$r = \sum_{i=1}^I e \quad (4.4)$$

Dörtgen sınır mesafesi hesaplandıktan sonra merkez noktanın x ve y koordinat değerleri ile toplama ve çıkarma işlemi yapılarak, sanal dörtgenin sınır koordinatları elde edilir.

Hesaplamalar da yer alan merkez noktanın koordinatları basisX, basisY ve dörtgenin sınır koordinatları maxX, maxY, minX, minY olarak aşağıda ki şekilde formüle dökülür;

$$maxX = basisX + r \quad (4.5)$$

$$maxY = basisY + r \quad (4.6)$$

$$minX = basisX - r \quad (4.7)$$

$$minY = basisY - r \quad (4.8)$$

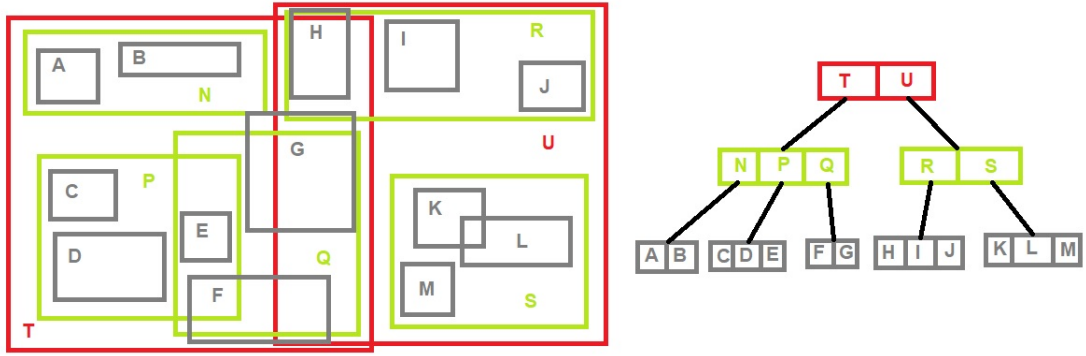
Dörtgenin sınırlarını oluşturan koordinat değerleri hesaplandıktan sonra alan içerisinde yer alan komşu nokta olup olmadığı kontrol edilir. Veri kümesinde yer alan x ve y koordinat değerleri dörtgenin koordinat değerleri ile kıyaslanır. Komşu noktaların x ve y eksenini koordinat kıyaslamaları matematiksel olarak şu şekilde ifade edilir;

$$(x < maxX \wedge x > minX) \wedge (y < maxY \wedge y > minY) \quad (4.9)$$

Alan içerisinde nokta tespit edilene kadar devam eden bu döngü sonucunda belirlenen noktalar için en yakın komşu mesafe fonksiyonları uygulanır, döngü belirlenen bir sayıda sonlandırılır ve herhangi bir komşu tespit edilmez ise işlem bu aşamada mesafe hesabı yapılmadan sonlanır.

4.3. r-Ağacı Algoritması

Uzamsal aramalarda alan taraması temelli çalışan bir algoritmadır. Ana çalışma mantığı veri kümesinde yer alan noktaları minimum boyutta dörtgenler ile yoğunluklu bulundukları alanlarda sınırlandırmaktır. Her dörtgen içerisinde yer alan eleman sayısı dengeli bir şekilde dağılmalıdır. Şekil 4.6'da görüldüğü gibi bu işlem veri kümesinin yoğunluğu ve boyutuna göre gerekli görüldüğü kadar tekrarlanabilir. Son durumda dengeli bir şekilde dağılmış veri kümesi ağaçları elde edilir.



Şekil 4.6. r-Ağacı çalışma yapısı

Diskte yer alan bir veri kümesi üzerinde çalışılıyor ise disk düğümleri ile disk bölümleri eşleşir. Minimum sayıda düğümde çalışması gereken bir uzamsal arama yapısı oluşturulmuş olur (Guttman, 1984).

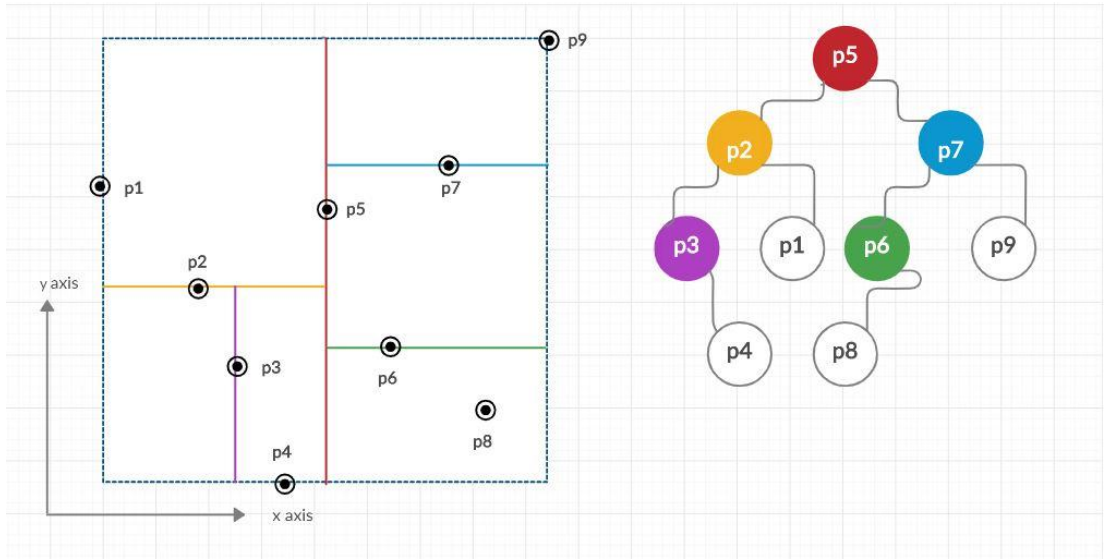
Veri kümesi bölümlendirilerek ağaç yapısı oluşturulduktan sonra merkez nokta etrafına sanal bir dörtgen oluşturulur. Merkez nokta etrafına oluşturulan dörtgenin sınırları değişkendir istenilen boyutlarda oluşturulabilir ancak verimli bir arama için en az sayıda düğümle çıkışacak dörtgen çizilmelidir. Bu dörtgenin sınırları ağaç yapısı, veri büyüklüğü ve veri alanına göre belirlenmelidir.

Sıradaki aşamada merkez nokta etrafına çizilen dörtgenin oluşturulan r-ağacı düğümlerindeki hangi dörtgenler ile çakıştığı tespit edilir. Tespit edilen düğümlerde dörtgenler olduğu için mesafe hesaplamaları sadece bu dörtgenlerin sınırları içerisinde yer alan komşular için uygulanır. Kesişen düğümler dışında ki alanda arama yapılmayacağı için kesişim dışı alan yok sayılır. Son aşamada kesişime dâhil olan alanlarda mesafe hesaplama fonksiyonları kullanılır.

4.4. kd-Ağacı Algoritması

İkili terimli bir ağaç algoritmasıdır. Her düğümü sağ ve sol olmak üzere iki alt düğüme ayrılır. Her döngü seviyesinde arama yapılacak alan bölerek yeni boyutta düğümler meydana getirir. Genellikle veri kümesinin orta noktası her döngüde merkez kabul edilerek o noktadan bölünme gerçekleşir.

Örneğin 2 boyutlu bir düzlemde bölümlendirmenin ilk aşaması olarak veri kümesinin x eksen koordinatlarının orta noktası alınır, y eksenine paralel olacak şekilde dikey bir şekilde sağ ve sol olarak bölümlendirilen veri kümesi 1. düğümü oluşturur. İkinci aşamada sağ ve sol olarak ayrılan küme kendi içlerinde yer alan veri kümelerine göre y eksen koordinatlarının orta noktasına bakılarak aşağı ve yukarı olarak bölümlendirilerek 2. düğümü oluştururlar. 3 aşamada tekrar her bir bölüm x eksenine göre bölümlendirilmelidir. Şekil 4.7’de görüldüğü gibi birinci ve ikinci aşamayı sırasıyla takip etmek koşulu ile bu işlem veri setinin büyüklüğü ve yoğunluğuna göre gereği kadar tekrar edebilmektedir.



Şekil 4.7. kd-Ağacı çalışma yapısı

Düğüm oluşturulduktan sonra merkez noktanın koordinatları ile hesaplama yapılır ve merkez noktanın hangi düğüm içerisinde yer aldığı tespit edilir. Belirlenen düğüm sınırları içerisinde ki komşu noktalardan en yakın komşu ya da komşuları tespit etmek için mesafe hesaplama fonksiyonları kullanılır. En yakın komşu “k” boyutsal (“d” dimensional) bölümlendirilmiş alanda kd-ağacı ile tespit edilmiş olur.

kd-Ağacı yöntemi için en yakın komşu hesaplaması yaparken 1 den fazla en yakın komşuya ihtiyaç olduğunda bu algoritmanın çoklu uygulanması gerekmektedir. Ancak çoklu uygulama ile birden fazla komşu hesaplanabilir (Hou vd., 2019).

5. UYGULAMA

Bu bölümde simülasyon çalışmaları ve çalışmalar sonucu oluşan veriler yer almaktadır.

5.1. Simülasyon Çalışmaları

Simülasyon çalışmaları Türkiye ülkesinin İstanbul şehrinde yer alan şehir bileşenlerinden taksi, taksi durağı, otopark noktaları ve eczaneler için oluşturulmuştur. Çalışmada test gerçekleştirilmiş durumların ana fikri bir alanda yer alan nokta yoğunluğu temellidir.

İstanbul şehrinde 17395 adet taksi ve 959 adet taksi durağı hizmet vermektedir (İstanbul Taksiciler Esnaf Odası, 2020). Şehirde 100000 araç park noktası yer almaktadır (İspark Kapasiteyi de Kaliteyi de Artırıyor, 2017). Elde edilen son verilere göre İstanbul şehrinde 5118 adet eczane hizmet vermektedir (Eczane Teknisyenleri Derneği, 2013). Şehir bileşenlerinin yoğunluğu; adetleri ve Çizelge 4.1'de yer alan İstanbul şehrinin sınır koordinatlarından elde edilen 2 boyutlu alan kullanılarak hesaplanır. Tüm testler Dünya koordinatlarının 2 boyutlu düzleme uygulanması üzerinden gerçekleşmiştir.

Çizelge 5.1. Boyutlu alan ve İstanbul şehri sınır koordinatları

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
2B Dünya Sınırları	180.0	90.0	-180.0	-90.0
İstanbul	41.58499	29.95724	40.802676	27.97643

İstanbul şehri koordinat sınırları incelendiğinde; x eksen koordinat farkı 1.98 birim y eksen koordinat farkı ise 0.782 birimdir. Koordinat sistemi üzerinde İstanbul şehri için 1.549 birim karelik alan oluşturmaktadır.

Şehrin koordinat sistemi üzerinde kapladığı alan ile içerisinde yer alan şehir bileşenleri tablodaki gösterildiği gibi bir birleri ile oranlanmıştır. Yaklaşık olarak aynı oranları sağlayan 4 büyük veri seti üzerinde en yakın şehir bileşenini tespit etmek için uzamsal arama algoritmaları kullanılacak ve performansları test edilmiştir.

Çizelge 5.2. İstanbul şehri 2B alan / şehir bileşeni oranları

	Şehir Bileşeni Adeti	Adet / Alan Oranı
Taksi	17395	11230
Taksi Durağı	950	613.3
Otopark Noktası	100000	64557.78
Eczane	5118	3304

Simülasyonlarda kullanılacak veri kümeleri csv dosya formatındadır. Dosyalarda 2B Dünya koordinat sisteminde yer alan konumlar yer almaktadır. Performansları karşılaştırılmış olan 6 algoritma python programlama dilinde kodlanmıştır. Büyük veri kümesinde gerçekleşen testler için intel xeon 2 sanal işlemcili, 24 gb bellek kapasitesine sahip Windows server 2019 işletim sistemine sahip bir bulut sanal makine kullanılmıştır. Simülasyonlar her durum için merkez nokta veri kümesinin içerisinde komşuların yoğun olduğu ve komşuların az yoğun olduğu bölgeden seçilerek 2 aşamada gerçekleştirilmiştir.

5.1.1. Taksi şehir bileşeni simülasyon çalışmaları

İstanbul şehrinde hizmet veren 17.395 taksi 1,549 birim karelik alanda 11.230 taksi / alan yoğunluğundadır. Testler de veri kümesi olarak 28,5127 birim karelik alanda yer alan 364371 nokta kullanılmıştır. Çizelge 5.3'te yer alan koordinat değerleri veri kümesinin sınırları olarak tanımlanacaktır.

Çizelge 5.3. Taksi şehir bileşeni veri kümesi sınır değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
Taksi Veri Kümesi	18.94081	59.84283	12.36515	55.50673

Çizelge 5.4'te görüldüğü üzere testlerde 2 adet merkez nokta kullanılmıştır. Noktalardan biri veri kümesinin yer aldığı alanın orta noktasından diğeri ise noktaların daha az yoğunlukta olduğu bir alanından seçilmiştir. Noktaların koordinatları arasında 1/100 değerinde değişiklikler gerçekleştiği için dörtgen sorgularda ihtiyacımız olan genişletme katsayısı 0,01 olarak kullanılmıştır.

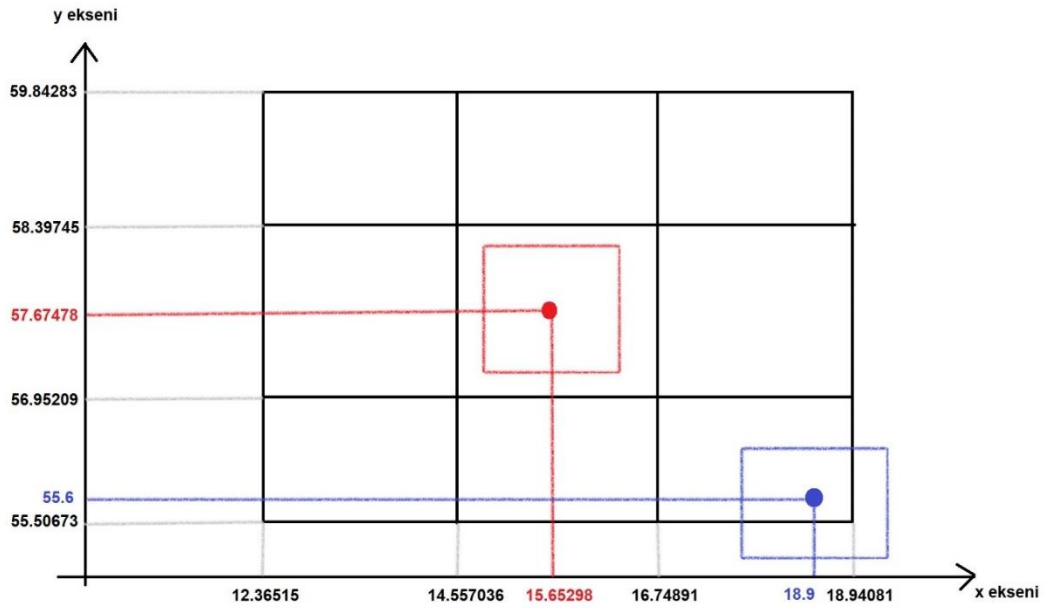
Çizelge 5.4. Taksi şehir bileşeni için merkez nokta değerleri

	x-axis coordinate	y-axis coordinate	e
Merkez Nokta 1	15.65298	57.67478	0.01
Merkez Nokta 2	18.9	55.6	0.01

28,5127 birim karelik veri seti alanı 9 adet düğüme bölümlendirilmiştir. r-ağacı algoritmasında kullanılacak olan bu düğümlerin koordinat sınırları ve merkez noktanın etrafına çizilecek olan sanal dörtgenin koordinatları Çizelge 5.5'te gösterildiği gibi tanımlanıp en yakın nokta araması öncesi programa kodlanmıştır. Şekil 5.1'de gösterildiği gibi merkez nokta etrafına çizilen sanal dörtgenin sınırları merkez noktanın koordinat sınırlarına 0,5 birim eklenip çıkartılarak hesaplanmıştır.

Çizelge 5.5. Taksi şehir bileşeni için r-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
r-Ağacı Düğüm1	14.557036	59.84283	12.36515	58.39745
r-Ağacı Düğüm2	16.74891	59.84283	14.557036	58.39745
r-Ağacı Düğüm3	18.94081	59.84283	16.74891	58.39745
r-Ağacı Düğüm4	14.557036	58.39745	12.36515	56.95209
r-Ağacı Düğüm5	16.74891	58.39745	14.557036	56.95209
r-Ağacı Düğüm6	18.94081	58.39745	16.74891	56.95209
r-Ağacı Düğüm7	14.557036	56.95209	12.36515	55.50673
r-Ağacı Düğüm8	16.74891	56.95209	14.557036	55.50673
r-Ağacı Düğüm9	18.94081	56.95209	16.74891	55.50673
Merkez Nokta 1 Dörtgeni	16.15298	58.17478	15.15298	57.17478
Merkez Nokta 2 Dörtgeni	19.4	56.1	18.4	55.1



Şekil 5.1. Taksi şehir bileşeni için r-ağacı simülasyon grafiği

Çizelge 5.6’da gösterildiği gibi kd-ağacı algoritmasının veri kümesinin alanını düğümlere bölümlendirmesi için ihtiyacı olan koordinat sınırları veri kümesinin maksimum ve minimum eksen değerlerini kapsayacak şekilde belirlenmiştir.

Çizelge 5.6. Taksi şehir bileşeni için kd-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
kd-ağacı	18.95	59.85	12.36	55.5

5.1.2. Taksi durağı şehir bileşeni simülasyon çalışmaları

İstanbul şehrinde hizmet veren 950 taksi durağı 1,549 birim karelik alanda 613,3 taksi durağı / alan yoğunluğundadır. Testler de veri kümesi olarak 2.427,3 birim karelik alanda yer alan 1.488.000 nokta kullanılmıştır. Çizelge 5.7’de yer alan koordinat değerleri veri kümesinin sınırları olarak tanımlanacaktır.

Çizelge 5.7. Taksi durağı şehir bileşeni veri kümesi sınır değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
Taksi Durağı Veri Kümesi	167.9928	-9.26485	96.82457	-43.4524

Çizelge 5.8’de görüldüğü üzere testlerde 2 adet merkez nokta kullanılmıştır. Noktalardan biri veri kümesinin yer aldığı alanın orta noktasından diğeri ise noktaların daha az yoğunlukta olduğu bir alanından seçilmiştir. Noktaların koordinatları arasında 1/1000 değerinde değişiklikler gerçekleştiği için dörtgen sorgularda ihtiyacımız olan genişletme katsayısı 0,001 olarak kullanılmıştır.

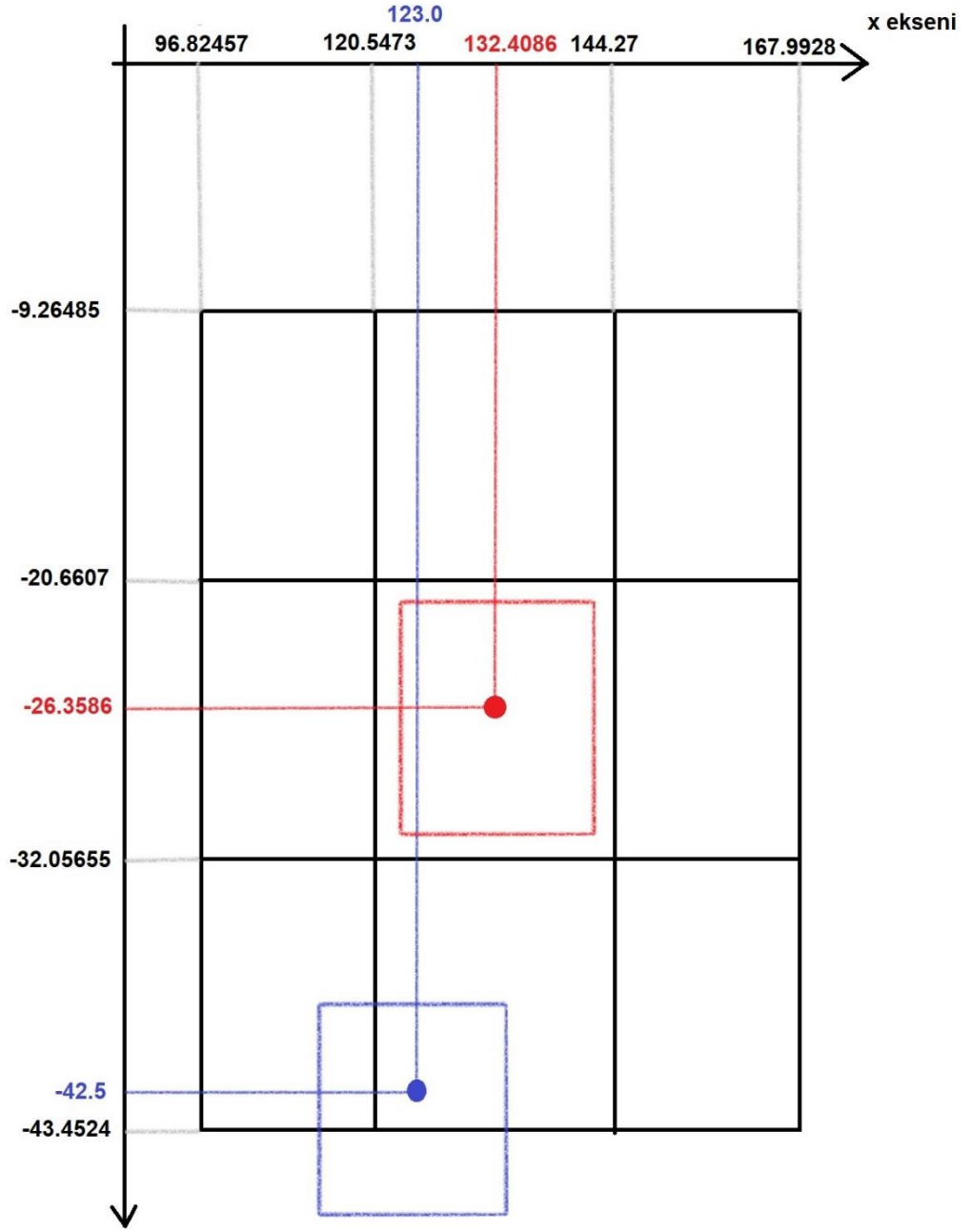
Çizelge 5.8. Taksi durağı şehir bileşeni için merkez nokta değerleri

	x-axis coordinate	y-axis coordinate	e
Merkez Nokta 1	132.4086	-26.3586	0.001
Merkez Nokta 2	123.0	-42.5	0.001

2.427,3 birim karelik veri seti alanı 9 adet düğüme bölümlendirilmiştir. r-ağacı algoritmasında kullanılacak olan bu düğümlerin koordinat sınırları ve merkez noktanın etrafına çizilecek olan sanal dörtgenin koordinatları Çizelge 5.9'da gösterildiği gibi tanımlanıp en yakın nokta araması öncesi programa kodlanmıştır. Şekil 5.2'de gösterildiği gibi merkez nokta etrafına çizilen sanal dörtgenin sınırları merkez noktanın koordinat sınırlarına 5 birim eklenip çıkartılarak hesaplanmıştır.

Çizelge 5.9. Taksi durağı şehir bileşeni için r-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
r-Ağacı Düğüm1	120.5473	-9.26485	96.82457	-20.6607
r-Ağacı Düğüm2	144.27	-9.26485	120.5473	-20.6607
r-Ağacı Düğüm3	167.9928	-9.26485	144.27	-20.6607
r-Ağacı Düğüm4	120.5473	-20.6607	96.82457	-32.05655
r-Ağacı Düğüm5	144.27	-20.6607	120.5473	-32.05655
r-Ağacı Düğüm6	167.9928	-20.6607	144.27	-32.05655
r-Ağacı Düğüm7	120.5473	-32.05655	96.82457	-43.4524
r-Ağacı Düğüm8	144.27	-32.05655	120.5473	-43.4524
r-Ağacı Düğüm9	167.9928	-32.05655	144.27	-43.4524
Merkez Nokta 1 Dörtgeni	137.4086	-21.3586	128.4086	-31.3586
Merkez Nokta 2 Dörtgeni	128	-37.5	118	-47.5



Şekil 5.2. Taksi durağı şehir bileşeni için r-ağacı simülasyon grafiği

Çizelge 5.10'da gösterildiği gibi kd-ağacı algoritmasının veri kümesinin alanını düğümlere bölümlendirmesi için ihtiyacı olan koordinat sınırları veri kümesinin maksimum ve minimum eksen değerlerini kapsayacak şekilde belirlenmiştir.

Çizelge 5.10. Taksi durağı şehir bileşeni için kd-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
kd-ağacı	168	-9.0	96.5	-44.5

5.1.3. Otopark noktası şehir bileşeni simülasyon çalışmaları

İstanbul şehrinde hizmet veren 100.000 otopark noktası 1,549 birim karelik alanda 64.557,7 otopark noktası / alan yoğunluğundadır. Testler de veri kümesi olarak 1,2082 birim karelik alanda yer alan 81.000 nokta kullanılmıştır. Çizelge 5.11’de yer alan koordinat değerleri veri kümesinin sınırları olarak tanımlanacaktır.

Çizelge 5.11. Otopark noktası şehir bileşeni veri kümesi sınır değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
Otopark Noktası Veri Kümesi	51.61665	26.14359	50.75345	24.74386

Çizelge 5.12’de görüldüğü üzere testlerde 2 adet merkez nokta kullanılmıştır. Noktalardan biri veri kümesinin yer aldığı alanın orta noktasından diğeri ise noktaların daha az yoğunlukta olduğu bir alanından seçilmiştir. Noktaların koordinatları arasında 1/1000 değerinde değişiklikler gerçekleştiği için dörtgen sorgularda ihtiyacımız olan genişletme katsayısı 0,001 olarak kullanılmıştır.

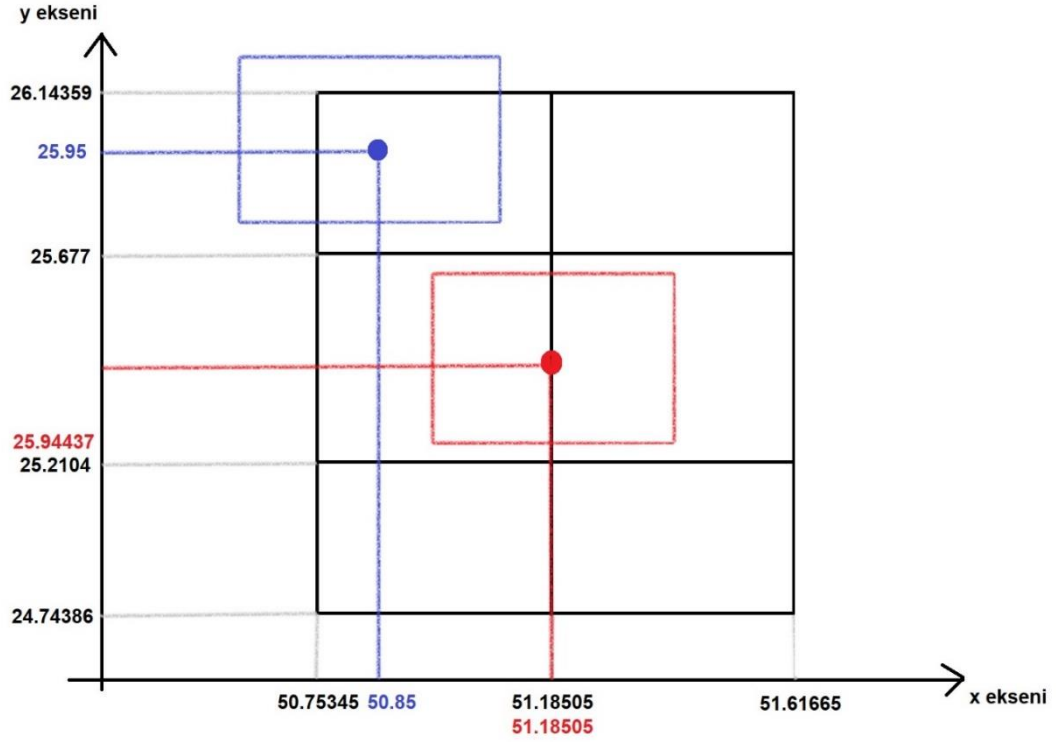
Çizelge 5.12. Otopark noktası şehir bileşeni için merkez nokta değerleri

	x-axis coordinate	y-axis coordinate	e
Merkez Nokta 1	51.185	25.4437	0.001
Merkez Nokta 2	50.85	25.95	0.001

1,2082 birim karelik veri seti alanı 6 adet düğüme bölümlendirilmiştir. r-ağacı algoritmasında kullanılacak olan bu düğümlerin koordinat sınırları ve merkez noktanın etrafına çizilecek olan sanal dörtgenin koordinatları Çizelge 5.13'te gösterildiği gibi tanımlanıp en yakın nokta araması öncesi programa kodlanmıştır. Şekil 5.3'te gösterildiği gibi merkez nokta etrafına çizilen sanal dörtgenin sınırları merkez noktanın koordinat sınırlarına 0,2 birim eklenip çıkartılarak hesaplanmıştır.

Çizelge 5.13. Otopark noktası şehir bileşeni için r-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
r-Ağacı Düğüm1	51.18505	26.14359	50.75345	25.677
r-Ağacı Düğüm2	51.61665	26.14359	51.18505	25.677
r-Ağacı Düğüm3	51.18505	25.677	50.75345	25.2104
r-Ağacı Düğüm4	51.61665	25.677	51.18505	25.2104
r-Ağacı Düğüm5	51.18505	25.2104	50.75345	24.74386
r-Ağacı Düğüm6	51.61665	25.2104	51.18505	24.74386
Merkez Nokta 1 Dörtgeni	51.385	25.6437	50.985	25.2437
Merkez Nokta 2 Dörtgeni	51.05	26.15	50.65	25.75



Şekil 5.3. Otopark noktası şehir bileşeni için r-ağacı simülasyon grafiği

Çizelge 5.14'te gösterildiği gibi kd-ağacı algoritmasının veri kümesinin alanını düğümlere bölümlendirmesi için ihtiyacı olan koordinat sınırları veri kümesinin maksimum ve minimum eksen değerlerini kapsayacak şekilde belirlenmiştir.

Çizelge 5.14. Otopark noktası şehir bileşeni için kd-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
kd-ağacı	51.65	26.15	50.75	24.74

5.1.4. Eczane şehir bileşeni simülasyon çalışmaları

İstanbul şehrinde hizmet veren 5.118 eczane 1,549 birim karelik alanda 3.304 eczane / alan yoğunluğundadır. Testler de veri kümesi olarak 33 birim karelik alanda yer alan 115.000 nokta kullanılmıştır. Çizelge 5.15'te yer alan koordinat değerleri veri kümesinin sınırları olarak tanımlanacaktır.

Çizelge 5.15. Eczane şehir bileşeni veri kümesi sınır değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
Eczane Veri Kümesi	-13.5125	66.56446	-24.5316	63.39966

Çizelge 5.16’da görüldüğü üzere testlerde 2 adet merkez nokta kullanılmıştır. Noktalardan biri veri kümesinin yer aldığı alanın orta noktasından diğeri ise noktaların daha az yoğunlukta olduğu bir alanından seçilmiştir. Noktaların koordinatları arasında 1/100 değerinde değişiklikler gerçekleştiği için dörtgen sorgularda ihtiyacımız olan genişletme katsayısı 0,01 olarak kullanılmıştır.

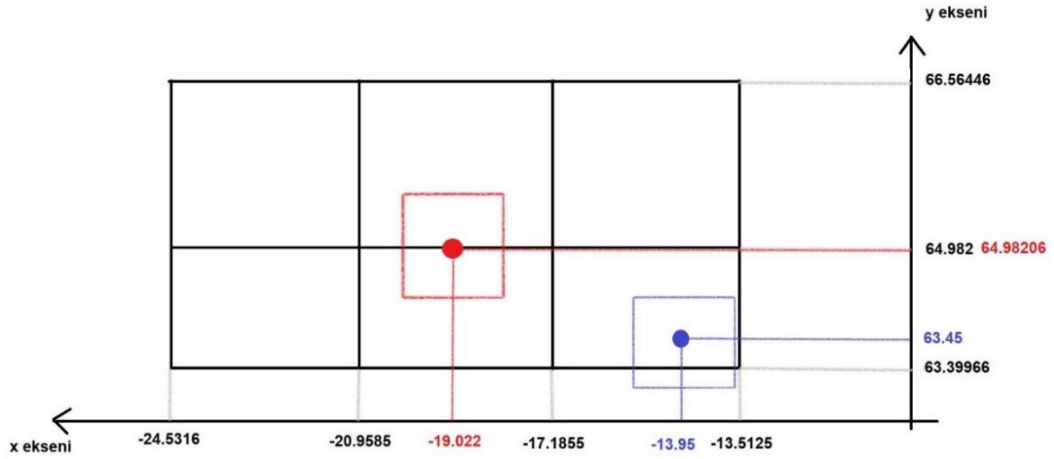
Çizelge 5.16. Eczane şehir bileşeni için merkez nokta değerleri

	x-axis coordinate	y-axis coordinate	e
Merkez Nokta 1	-19.022	64.98206	0.01
Merkez Nokta 2	-13.95	63.45	0.01

33 birim karelik veri seti alanı 6 adet düğüme bölümlendirilmiştir. r-ağacı algoritmasında kullanılacak olan bu düğümlerin koordinat sınırları ve merkez noktanın etrafına çizilecek olan sanal dörtgenin koordinatları Çizelge 5.17’de gösterildiği gibi tanımlanıp en yakın nokta araması öncesi programa kodlanmıştır. Şekil 5.4’te gösterildiği gibi merkez nokta etrafına çizilen sanal dörtgenin sınırları merkez noktanın koordinat sınırlarına 0,4 birim eklenip çıkartılarak hesaplanmıştır.

Çizelge 5.17. Eczane şehir bileşeni için r-ağacı koordinat değerleri

	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
r-Ağacı Düğüm1	-20.9585	66.56446	-24.5316	64.982
r-Ağacı Düğüm2	-17.1855	66.56446	-20.9585	64.982
r-Ağacı Düğüm3	-13.5125	66.56446	-17.1855	64.982
r-Ağacı Düğüm4	-20.9585	64.982	-24.5316	63.39966
r-Ağacı Düğüm5	-17.1855	64.982	-20.9585	63.39966
r-Ağacı Düğüm6	-13.5125	64.982	-17.1855	63.39966
Merkez Nokta 1 Dörtgeni	-18.622	65.38206	-19.422	64.58206
Merkez Nokta 2 Dörtgeni	-13.55	63.85	-14.35	63.05



Şekil 5.4. Eczane şehir bileşeni için r-ağacı simülasyon grafiği

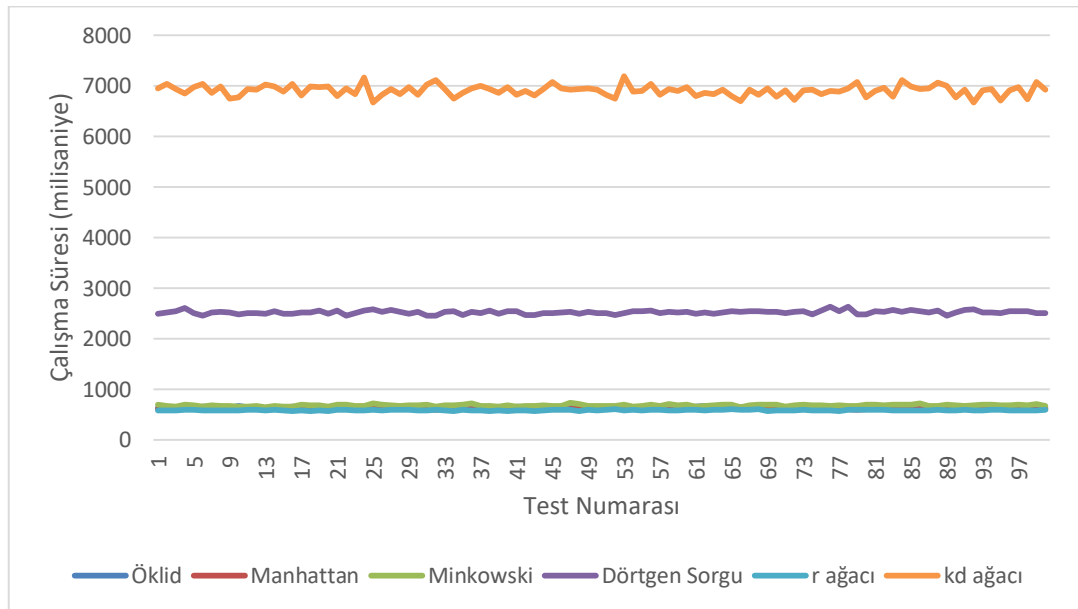
Çizelge 5.18’de gösterildiği gibi kd-ağacı algoritmasının veri kümesinin alanını düğümlere bölümlendirmesi için ihtiyacı olan koordinat sınırları veri kümesinin maksimum ve minimum eksen değerlerini kapsayacak şekilde belirlenmiştir.

Çizelge 5.18. Eczane şehir bileşeni için kd-ağacı koordinat değerleri

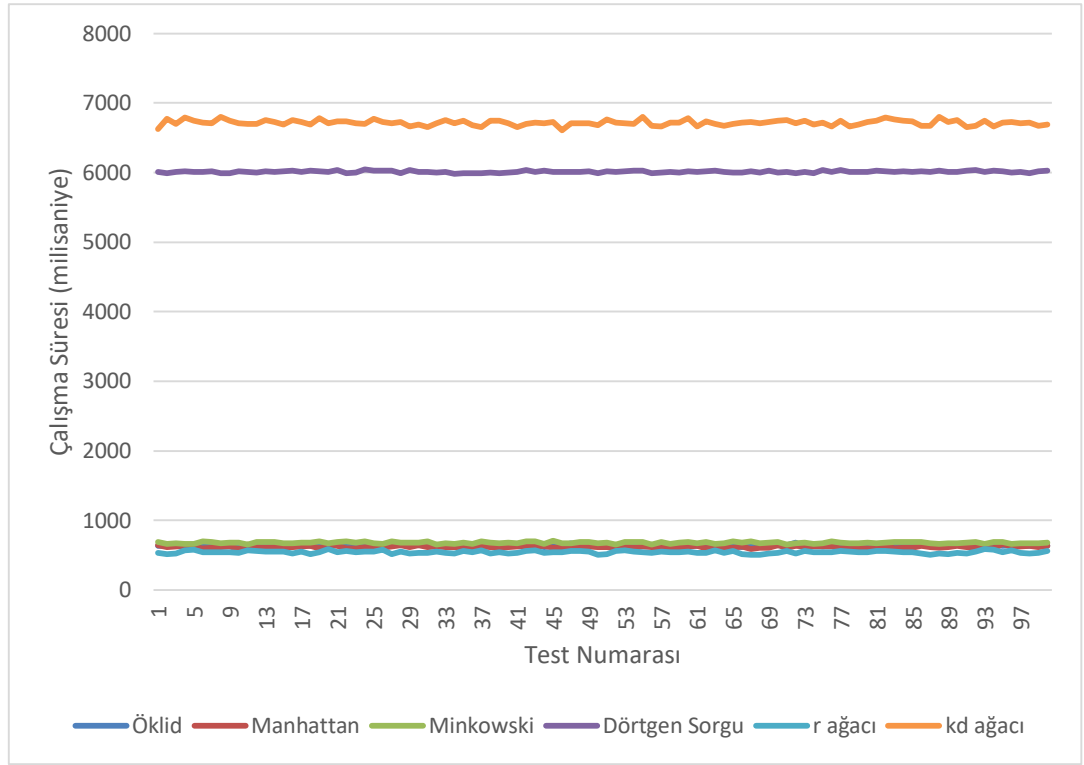
	Maksimum x-ekseni değeri	Maksimum y-ekseni değeri	Minimum x-ekseni değeri	Minimum y-ekseni değeri
kd-ağacı	-13.5	66.6	-24.6	63.3

5.2. Simülasyon Çalışmaları Bulguları

Taksi şehir bileşeni için oluşturulan simülasyon çalışması her bir algoritma ve nokta için 100 kere test edildi. Veri kümesine dâhil edilen ve en yakın komşusu bulunacak olan her iki nokta içinde %100 oranında doğru sonuç verdi. Merkez nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 643,29 ms, Manhattan fonksiyonu 617, 60 ms, Minkowski algoritması 681,64 ms, Dörtgen sorgu algoritması 2524,53 ms, r-ağacı algoritması 586,73 ms ve kd-ağacı 6912,27 ms süreleri elde edilmiştir. Az yoğun aladan seçilen nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 640,96 ms, Manhattan fonksiyonu 617,25 ms, Minkowski algoritması 678,27 ms, Dörtgen sorgu algoritması 6012,16 ms, r-ağacı algoritması 542,65 ms ve kd-ağacı 6716,34 ms süreleri elde edilmiştir.

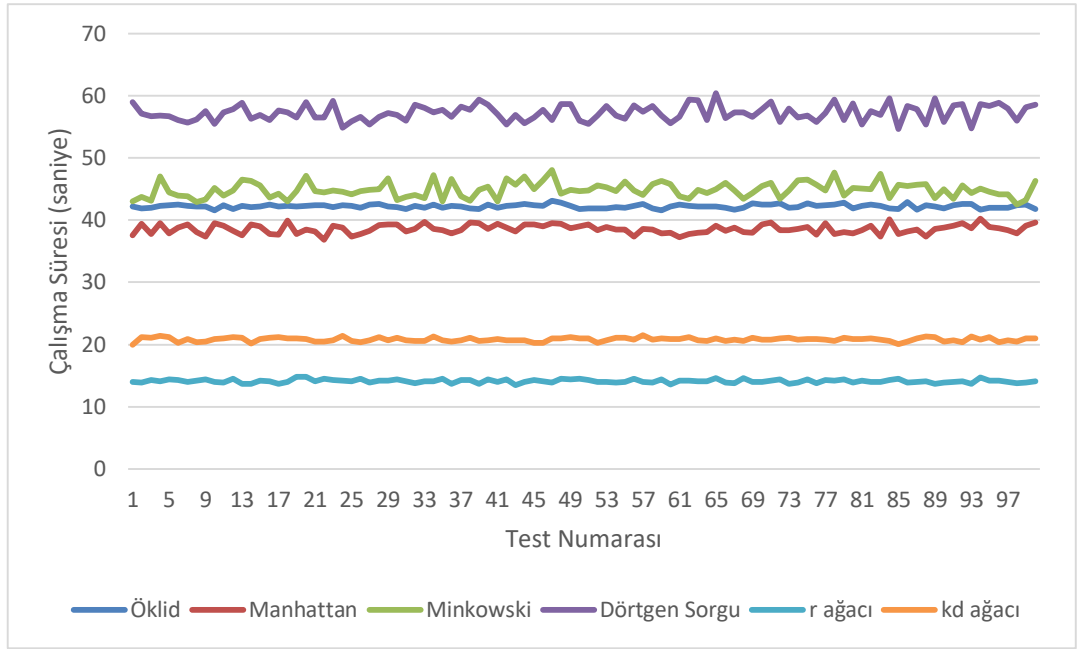


Şekil 5.5. Taksi şehir bileşeni merkez nokta 1 için test süreleri

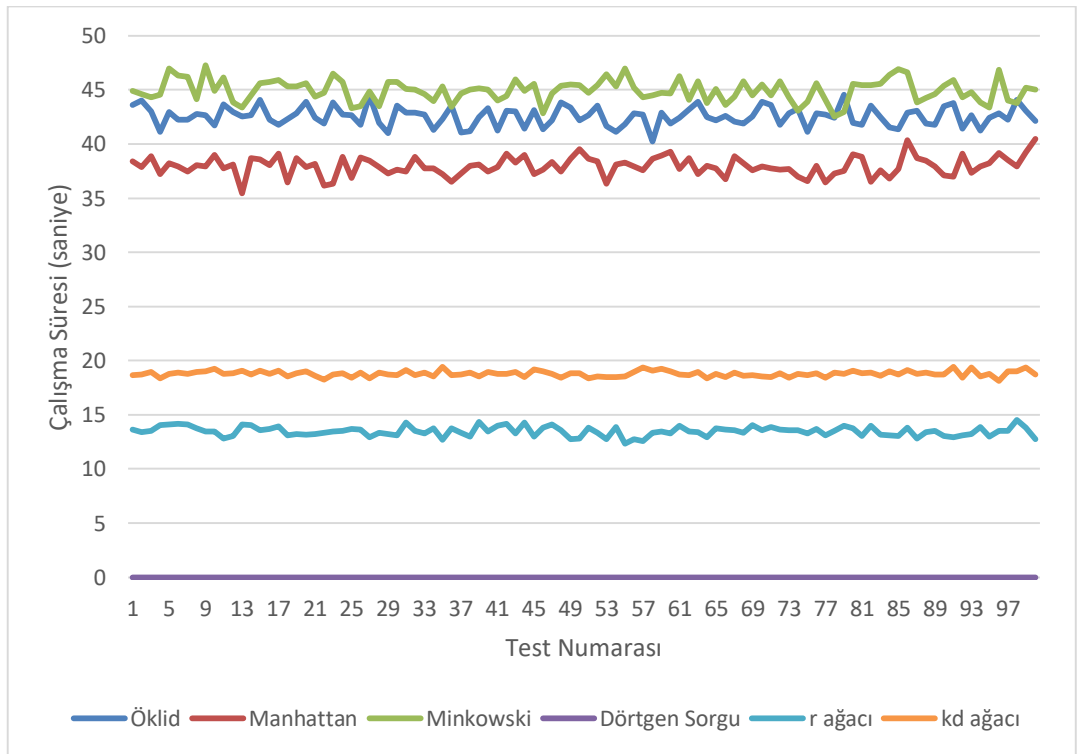


Şekil 5.6. Taksi şehir bileşeni merkez nokta 2 için test süreleri

Taksi durağı şehir bileşeni için oluşturulan simülasyon çalışması her bir algoritma ve nokta için 100 kere test edildi. Veri kümesine dahil edilen ve en yakın komşusu bulunacak olan merkez nokta içinde %100 oranında doğru sonuç verdi. Az yoğun alandan seçilen nokta için yapılan testlerde dörtgen sorgular sorgu limitini aşmıştır ve bir sonuca ulaşılmamıştır. Merkez nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 42,21 sn, Manhattan fonksiyonu 38,57 sn, Minkowski algoritması 44,86 sn, Dörtgen sorgu algoritması 57,22 sn, r-ağacı algoritması 14,11 sn ve kd-ağacı 20,80 sn süreleri elde edilmiştir. Az yoğun aladan seçilen nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 42,55 sn, Manhattan fonksiyonu 37,96 sn, Minkowski algoritması 44,94 sn, Dörtgen sorgu algoritması için çalışma süresi olmadığı için ortalama süre hesaplanmamıştır, r-ağacı algoritması 13,46 sn ve kd-ağacı 18,78 sn süreleri elde edilmiştir.



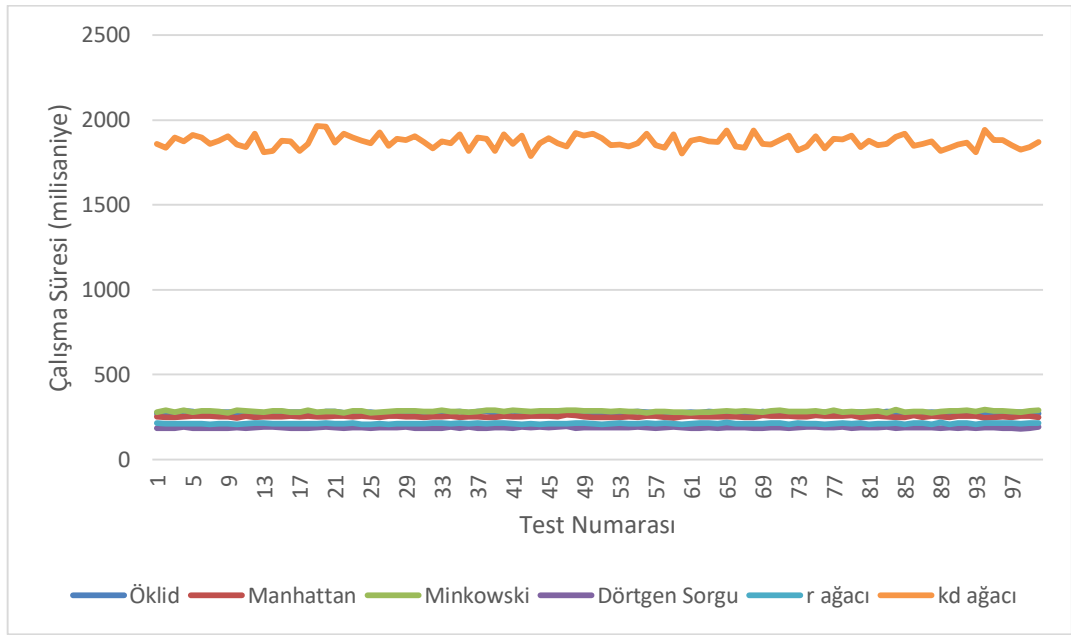
Şekil 5.7. Taksi durağı şehir bileşeni merkez nokta 1 için test süreleri



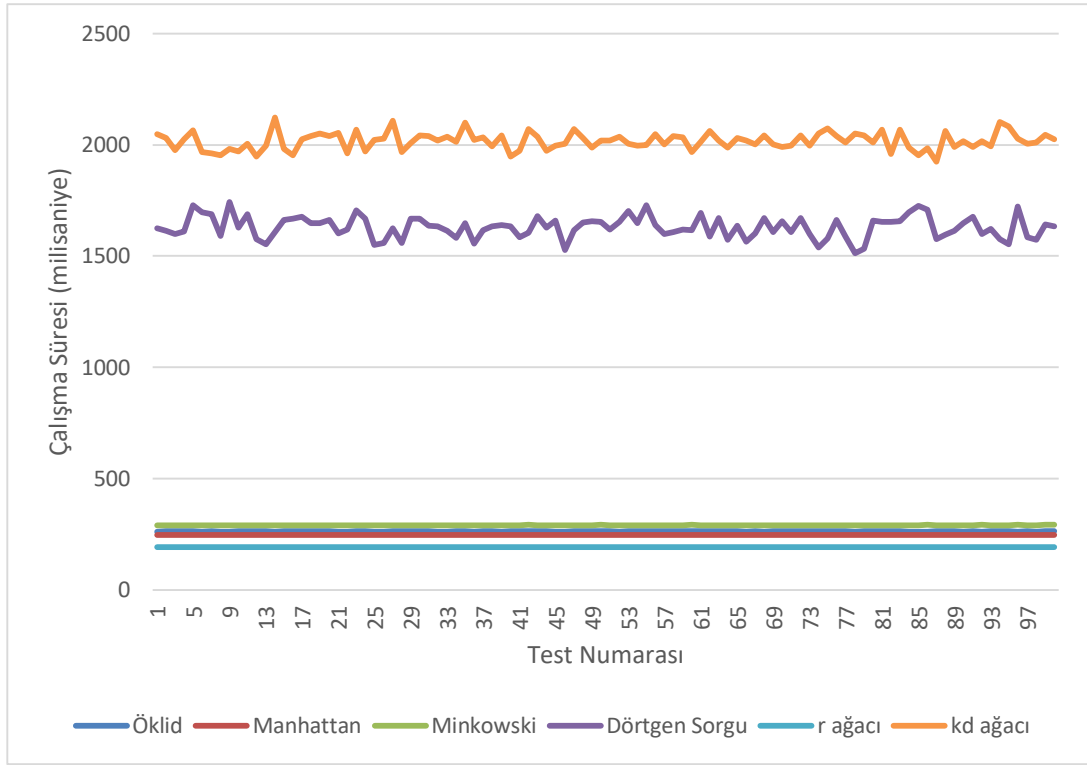
Şekil 5.8. Taksi durağı şehir bileşeni merkez nokta 2 için test süreleri

Otopark noktası şehir bileşeni için oluşturulan simülasyon çalışması her bir algoritma ve nokta için 100 kere test edildi. Veri kümesine dahil edilen ve en yakın komşusu bulunacak olan her iki nokta içinde %100 oranında doğru sonuç

verdi. Merkez nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 277,35 ms, Manhattan fonksiyonu 253,13 ms, Minkowski algoritması 283,82 ms, Dörtgen sorgu algoritması 187,36 ms, r-ağacı algoritması 211,84 ms ve kd-ağacı 1872,54 ms süreleri elde edilmiştir. Az yoğun aladan seçilen nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 263,00 ms, Manhattan fonksiyonu 246,98 ms, Minkowski algoritması 291,02 ms, Dörtgen sorgu algoritması 1629,10 ms, r-ağacı algoritması 193,02 ms ve kd-ağacı 2016,76 ms süreleri elde edilmiştir.

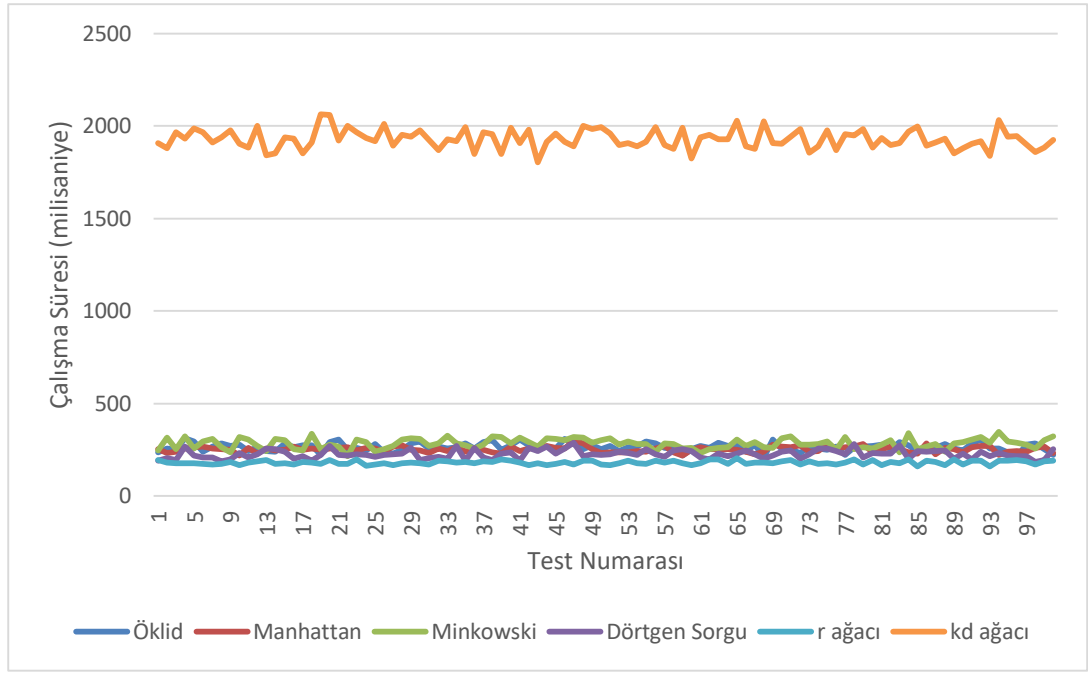


Şekil 5.9. Otopark noktası şehir bileşeni merkez nokta 1 için test süreleri

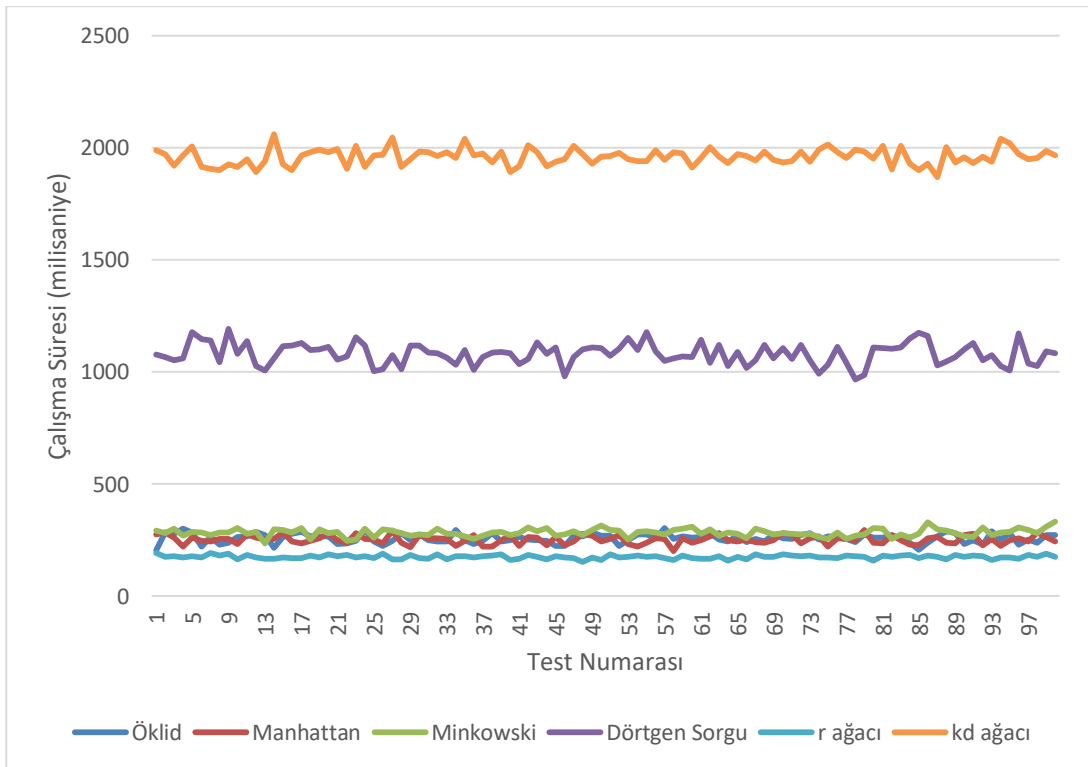


Şekil 5.10. Otopark noktası şehir bileşeni merkez nokta 2 için test süreleri

Eczane şehir bileşeni için oluşturulan simülasyon çalışması her bir algoritma ve nokta için 100 kere test edildi. Veri kümesine dahil edilen ve en yakın komşusu bulunacak olan her iki nokta içinde %100 oranında doğru sonuç verdi. Merkez nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 263,22 ms, Manhattan fonksiyonu 251,62 ms, Minkowski algoritması 282,88 ms, Dörtgen sorgu algoritması 226,12 ms, r-ağacı algoritması 180,41 ms ve kd-ağacı 1929,24 ms süreleri elde edilmiştir. Az yoğun aladan seçilen nokta için Şekilde yer alan algoritma çalışma sürelerinin ortalaması alındığında; Öklid fonksiyonu 258,33 ms, Manhattan fonksiyonu 252,05 ms, Minkowski algoritması 283,01 ms, Dörtgen sorgu algoritması 1081,06 ms, r-ağacı algoritması 175,04 ms ve kd-ağacı 1959,81 ms süreleri elde edilmiştir.



Şekil 5.11. Eczane şehir bileşeni merkez nokta 1 için test süreleri



Şekil 5.12. Eczane şehir bileşeni merkez nokta 2 için test süreleri

6. SONUÇ VE ÖNERİLER

Yapılan Kentsel Sistemler ve Ulaştırma Yönetimi Yüksek Lisans Programı bünyesinde hazırlanan “Dinamik Parametrelili Uzaysal Arama İle Uygun Düğüm Seçimi Algoritmalarının Simölasyonu ve Karışılaltırılması” başlıklı tez çalışmasında dijitalleşen ve gün geçtikçe akıllı hale gelen şehirlerde akıllı şehir bileşenlerinin bulunması için kullanılan mobil uygulamaların performansını etkileyen algoritmaların performansları değeriendirilmiştir.

Algoritmaların performans değeriendirmeleri için yapılan 4 farklı şehir bileşeni modelinde 6 farklı algoritma test edilmiştir. Her şehir bileşeni için her bir algoritma 100 defa uygulanmıştır.

Taksi şehir bileşeni için yapılan testlerde veri kümesinin merkezinden seçilen bir nokta için r-ağacı en hızlı sonuca ulaşan algoritma olmuştur. Veri kümesinin az yoğun olduğu bir bölgeden seçilen nokta da ise en iyi performansı tekrar r-ağacı algoritması sergilemiştir. Merkez noktada en kötü performansı kd-ağacı, az yoğun bölgeden seçilen noktada ise kd-ağacı ile dörtgen sorgular sergilemişlerdir.

Taksi durağı şehir bileşeni için yapılan testlerde veri kümesinin merkezinden seçilen bir nokta r-ağacı ve kd-ağacı diğer algoritmalara göre daha iyi performans göstermişlerdir. Veri kümesinin az yoğun bölgesinden seçilen noktada ise sonuç değışmemiş r-ağacı ve kd -ağacı iyi performans göstermişlerdir. Bu simölasyonda dörtgen sorgular merkez nokta durumunda en kötü performansı gösterirken, veri kümesinin az yoğun bölgesinden seçilen noktada ise limit sorgu süresi ve adeti aşılaraq bir sonuca ulaşamamıştır.

Otopark noktası şehir bileşeni için yapılan testlerde veri kümesinin merkezinden seçilen nokta için dörtgen sorgular iyi performans gösterirken, veri kümesinin az yoğun bölgesinden seçilen noktada r-ağacı iyi performans göstermiştir. Merkez nokta için kötü performans sergileyen algoritma kd-ağacı olurken, az yoğun bölgeden seçilen nokta için kd-ağacı ve dörtgen sorgular kötü performans

sergilemişlerdir. Dörtgen sorgular veri kümesinin yoğun ve homojen olduğu veri setleri için önerilebilir ancak şehir bileşenleri aramalarında homojen dağılım zor görüntüleneceği için tek başına kullanılması önerilmez.

Eczane şehir bileşeni için yapılan testlerde merkez nokta için r-ağacı iyi performans göstermiştir. Az yoğun bölgeden seçilen noktada ise r-ağacı iyi performansını devam ettirmiştir. Her iki seçilen noktada kd-ağacı kötü performans göstermiştir. Az yoğun bölgeden seçilen noktada dörtgen sorgu algoritması performans kaybına uğramıştır.

Bu dört simülasyon kapsamında algoritmaları değerlendirecek olursak; bir alanda nokta yoğunluğu yüksek ise dörtgen sorgular tercih edilebilmektedir. Ancak aynı veri modelinin az yoğun bölümünden bir arama gerçekleştiğinde dörtgen sorgu algoritmasında ciddi performans kayıpları yaşanmaktadır. Nokta yoğunluğunun az olduğu aramalarda r-ağacı ve kd-ağacı iyi performanslarından dolayı tercih edilebilir. Arama yapılacak noktanın merkezden veya az yoğun bölgeden seçilmiş olması nokta yoğunluğu az olduğunda fark etmemektedir. Ağaç algoritmaları performanslarını devam ettirmektedirler.

Arama yapılacak olan noktanın veri kümesinin herhangi bir noktasından seçilmiş olması k-en yakın komşu algoritması kapsamında teste tabi tutulan Öklid, Manhattan ve Minkowski fonksiyonları için bir fark yaratmamıştır. Bunun sebebi arama noktası nerede olursa olsun veri kümesindeki tüm noktalar için hesaplama yapılacak olmasıdır. Bu nedenle arama noktası değiştirmek herhangi bir performans değişimine sebep olmamıştır ancak veri kümesinin büyümesi performanslarını ciddi biçimde düşürmüştür. Büyük veri aramalarında tek başlarına kullanılmaları önerilmemektedir.

Simülasyon çalışmalarında yapılan testlerin genelinde r-ağacı iyi performansı ile ön plana çıkmıştır. Arama yapılacak noktanın merkezden az yoğun bölgeye taşınması performansını olumlu etkilemiştir. Bunun nedeni ise r-ağacında arama yapılacak noktanın etrafına sanal bir dörtgen çizilip bu dörtgenin önceden sınırları programa kodlanan düğümler ile karşılaştırılmasından kaynaklıdır. Arama

alanı belirli olan uygulamalarda düğüm sınırları tekrar tekrar hesaplanmadığı için r-ağacı sadece kesişen düğümlerde arama yaparak arama alanını düşürmüş ve iyi performans göstermiştir. Özellikle arama alanı belirli ve kısa vadede değişmeyen şehir bileşenleri gibi büyük veri modelleri için önerilen uzamsal arama algoritmasıdır.

Çalışmalarda yer alan büyük veri setlerinde kötü performansı ile ön plana çıkan kd-ağacı sadece az yoğunluklu veride iyi performans sergilemiştir. Bunun nedeni ise her ne kadar bir ağaç algoritması olsa da kd-ağacı düğümleri hesaplarken veri boyutuna göre işlem yapmaktadır. Alanda yer alan nokta ne kadar fazla ise kd-ağacı performansı o kadar düşmektedir. Her test aşamasından tekrar düğüm hesaplaması yapması performansını etkileyen büyük bir faktördür. Sınırları belirli bir alanda özellikle veri yoğunluğu yaratan şehir bileşenleri için kullanılması önerilmemektedir.

KAYNAKLAR

- Caragliu vd., 2009. Smart Cities in Europe. Journal of Urban Technology, 18, 65-82.
- Cover, T., Hart, P., 1967. Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13, 21-27.
- Dierbach, C., 2012. Introduction to Computer Science Using Python: A Computational Problem-solving Focus, Wiley Publishing.
- Doukas, G., Jupyter Kernels, 2020. EriřimTarihi: 01.05.2020. <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
- Eczane Teknisyenleri Derneęi, 2013. İllerde Eczane sayısı, Eriřim tarihi: 05.05.2020. <https://eczaneteknisyenleridernegi.com/wp-content/uploads/2013/07/illerde-eczane-sayisi.pdf>
- Guttman, A., 1984. R Trees: A Dynamic Index Structure for Spatial Searching. Sigmod Record, 14, 47-57.
- Hou, W., Li, D., Xu, C., Zhang, H., Li, T., 2019. An Advanced k Nearest Neighbor Classification Algorithm Based on KD-tree. 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), 10-12 Dec. 2018, Chongqing, China.
- Hu, L., Huang, M., Ke, S., Tsai, C., 2016. The Distance Function Effect on k-nearest Neighbor Classification for Medical Datasets. SpringerPlus, 5, 1304.
- Issycom, 2007, Eriřim tarihi: 03.05.2020. <https://www.issy.com/issygrid>
- İBB Eczane Mobil Uygulaması, 2020, Eriřim Tarihi: 03.05.2020. <https://cbsakademi.ibb.istanbul/proje/ibb-eczane-mobil-uygulamasi/>
- Internet World StatsUsage and Population Statistics, 2020. Eriřim tarihi: 01.05.2020. <https://www.internetworldstats.com/emarketing.htm#links>
- İsbike Nasıl Çalışır?, 2020, Eriřim Tarihi: 03.05.2020. <https://www.isbike.istanbul/nasilcalisir>
- İspark, 2017. İstanbul Kapasiteyi de Kaliteyi de Artırıyor, Eriřim Tarihi:05.05.2020. <https://ispark.istanbul/ispark-kapasiteyi-de-kaliteyi-de-artiriyor/>
- İstanbul Taksiciler Esnaf Odası, 2020. Eriřim Tarihi: 05.05.2020. <http://iteo.org.tr>
- iTaksi,2020, Eriřim tarihi:03.05.2020. <https://www.itaksi.com/#Neden-iTaksi>

- Kramers vd., 2014. Smart Sustainable Cities – Exploring ICT Solutions for Reduced Energy Use in Cities. *Environmental Modelling & Software*, 56, 52-62.
- Kurt, H., 2006. Göç Eğilimleri Ve Olası Etkileri, *Yönetim Bilimleri Dergisi*, 6(4), 148-178.
- Li, G., Tang, J., 2010. A New K-NN Query Algorithm Based on the Traversal and Search of the Dynamic Rectangle. 2010 International Conference on E-Product E-Service and E-Entertainment, 7-9 Nov. 2010, Henan, China.
- Line, 2020. Erişim Tarihi:03.05.2020, <https://linecorp.com/en/pr/news/en/2015/904>
- Liu D, Lim E, Ng W., 2002. Efficient k Nearest Neighbor Queries on Remote Spatial Databases Using range Estimation. In: *Proceedings of International Conference on Scientific and Statistical Databasesmanagement (SSDMB)*, 24–26 July 2002, 121–130, Edinburgh.
- Lyft Inc., 2020. Erişim Tarihi:03.05.2020. <https://www.lyft.com/driver/why-drive-with-lyft>
- Manville vd., 2014. Mapping Smart Cities in the EU. Directorate-General for Internal Policies, Policy Department A: Economic and Scientific Policy, 01 January 2014, 1,1-200.
- Ola, 2020. Erişim Tarihi:03.05.2020. <https://corporate.olacabs.com/#/how>
- Oliphant, T. E., 2007. Python for Scientific Computing. *Computing in Science & Engineering*, 9, 10-20.
- Şeker B., T., 2005. Bilgi Teknolojilerindeki Gelişmeler Çerçevesinde Bilgiye Erişimin Yeni Boyutları, *Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*,13.
- Taşçı, E., Onan, A., 2017. K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi., Erişim Tarihi: 01.06.2020, <https://www.semanticscholar.org/paper/K-En-Yak%C4%B1n-Koms%CC%A7u-Algoritmas%C4%B1-Parametrelerinin-Taschi-Onan/d62b68ab6395b6beef20caa1bf530ecf188f8dc8#citing-papers>
- T. C. Kalkınma Bakanlığı, 2013. Onuncu Kalkınma Planı (2014-2018), Kalkınma Bakanlığı Yayını, Ankara, Erişim Tarihi: 03.05.2020. <http://www.kalkinma.gov.tr>
- Tratt, L., 2009. Dynamically Typed Languages, *Advances in Computers*, 77, 149-184.

- Uber, 2009. Erişim Tarihi:03.05.2020. <https://www.uber.com/tr/tr/about/>
- We are Social, 2020. Erişim tarihi: 01.05.2020. <https://wearesocial.com/digital-2020>
- World Urbanization Prospects The 2014 Revision, 2014. Erişim tarihi: 02.05.2020.
<https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Report.pdf>
- Yovanof, G.S., Hazapis, G.N., 2009. An Architectural Framework and Enabling Wireless Technologies for Digital Cities & Intelligent Urban Environments, *Wireless Personal Communications*, 49, 445-463.
- Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R., 2017. Efficient kNN Classification with Different Numbers of Nearest Neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29, 1774-1785.
- Zhang, Z., Zhang, J., Yang, J., Yang, Y., 2007. A New Approach to Creating Spatial Index with R-Tree. 2007 International Conference on Machine Learning and Cybernetics, 19-22 Aug. 2007, Hong Kong, China.
- Zhu, Q., Gong, J., Zhang, Y., 2007. An Efficient 3D R-tree Spatial Index Method for Virtual Geographic Environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62, 217-224.

ÖZGEÇMİŞ

Adı Soyadı : Mert Can GİRĞİN
Doğum Yeri ve Yılı : EDİRNE, 27.02.1992
Medeni Hali : Evli
Yabancı Dili : İngilizce
E-posta : girgin.mert@yandex.com



Eğitim Durumu

Lise : Kâğıthane Anadolu Lisesi, 2010
Lisans : Doğu Akdeniz Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 2016
Yüksek Lisans : İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Kentsel Sistemler ve Ulaştırma Yönetimi Anabilim Dalı, 2020

Mesleki Deneyim

Esim Yazılım	2016-2017
İsbak A.Ş.	2017-...(devam ediyor)

Yayınları

Girgin M., Boyacı A., 2020. Performance Comparison of Spatial Search Algorithms for Specific Data Sets In Smart Cities, Journal of Technologies and Applied Sciences, In Press.